A Single Interface Remote Administration Tool
(SIRAT)

CSCI 8960
Thesis Equivalent Project in Computer Science

Submitted to the
Computer Science Department
University of Nebraska at Omaha

Mentoring

Chaired by
Dr. Azad Azadmanesh, Professor at University of Nebraska, Department of Computer Science. **azad@unomaha.edu**

Supervisory Committee

Dr. Parvathi Chundi, Professor at University of Nebraska, Department of Computer Science. **pchundi@mail.unomaha.edu**
Dr. Deepak Khazanchi, Professor at University of Nebraska, Department of ISQA. **khazanchi@unomaha.edu**

Submitted by

Rajesh Ratnala
July 2011

Table of Contents

1. Statement of Problem Area 4
1.1 Birth of the Idea and its Incentives 5
2 SIRAT Functional Specifications 6
2.1 Functions Performed 6
2.2.1 Windows Service 6
2.2.2 Windows Batch Files 6
2.2 User Interface Design 7
2.3 Integration with other Technologies 7
2.4 Error Conditions and System Messages 8
2.5 Additional Functionalities 8
2.6 Help Material and User Manual 8
2.7 Limitations and Restrictions 8
3 SIRAT Performance Requirements 9
3.1 Efficiency 9
3.2 Description of Reliability Measures 9
3.3 Error/Failure Detection 10
3.4 Security 10
3.5 Maintainability 10
4 SIRAT Design Overview 10
4.1 SIRAT Design Architecture 10
4.2 SIRAT User Interface Terminology 13
4.3 SIRAT Data Flow Diagram 13
4.4 SIRAT Database Tables 15
5 SIRAT Installation and Configuration Overview 15
5.1 Software Packages used in building SIRAT 15
5.2 Installations 16
5.3 Configuration 16
5.3.1 Configure MySQL 5.5.9 16
5.3.2 Configure JBoss Application Server 18
5.3.3 Windows Batch File Customization 21
6 SIRAT User Interface Overview 22
6.1 Login Page 23
6.2 Register Page 23
6.3 Home Page 24
6.4 Project List Page 24
6.5 Create Service Page 25
6.6 Reachability Page 26
6.7 About and Log Out Pages 27
7 SIRAT Verification 28
7.1 Test Bed Environment 28
7.1.1 Simple Home Network 28

7.1.2 Office Network 30
7.2 Items/Functions to be Tested 31
7.3 Description of Test Cases 31
7.4 Test Results 36
7.5.1 Login and Log out functionality 36
7.5.2 Registration 36
7.5.3 Create Project and Service 37
7.5.4 Run a Windows Service with Correct Information (Query/Start/Stop) 37
7.5.5 Run a Windows Service with Incorrect Information 39
7.5.6 Run a Windows Batch File with correct Information 39
7.5.7 Run a Windows Batch File with Incorrect Information 40
7.5.8 Reachability Test 41
7.6 Example: SIRAT Integration with Oracle Weblogic Application Server 47
7.7 Performance Testing 55
8 Conclusions 57
8.1 Summary 57
8.2 Problems Encountered and Solved 57
8.3 Suggestions for Future Extensions to Project 58
Appendix A: Remote Administration 60
Appendix B: Required Support Software to build SIRAT 60
Appendix C: PsTools Suite 61
Appendix D: JBoss Application Server 65
Appendix E: JRockit JVM 66
Appendix F: MySQL Database 69
Appendix G: Oracle Weblogic Application Server 70
Appendix H: MySQL 5.5.9 Installation 73
Appendix I: Installing the JRockit 1.6.0_20 81
Appendix J: Installing the JBoss 5.1.0 GA 85
Appendix K: Installing PSTools Suite 88
9 References 89

1. Statement of Problem Area

Remote administration refers to any method of controlling a computer from a remote location. Remote administration is becoming increasingly common and is often used when it is difficult or impractical to be physically near a system in order to use it, or in order to access web material that is not available in one's location. Any computer with an Internet connection, TCP/IP or on a Local Area Network (LAN) can be remotely administered. Remote administration can be used for any cluster of activities and can span multiple categories of servers, such as database servers, middleware servers, etc.

In general, a *middleware* is an interface that mediates between the application programs and the computing system. The application programs are often referred to as users or clients requesting services and the computing system is the entity providing the services. In an analogy, this is similar to client/server systems where the middleware acts as glue or the conduit to provide communication between the clients and the server. Middleware is also similar to the middle layer of three-tier single system architecture, except that it is stretched across multiple systems or applications. Some examples of middleware server technologies are JBoss and IBM Websphere.

With many administrators in the IT world, each one may have different needs and daily tasks. There may be some tasks that an administrator has to repeat daily on a frequent basis – this becomes a time consuming and unproductive activity. To quote as an example, a server administrator might need to start and stop the JBoss servers several times a day. If there were one or two servers, then this would be a simple task. What if there were a number of servers, like 30 or 40 servers that need to be started at different times? Connecting to each of these disparate boxes[1] individually and performing similar type of administrative activities repetitively are very time consuming and reduce a person's and thereby a firm's productivity significantly.

There exist many tools in the market with the capability of remotely administrating tasks on these servers. These cheap and inexpensive tools help a middleware administrator with day to day work by logging into separate interfaces in order to complete the tasks needed. The tools can constantly monitor these machines and servers and report any problem if they see any suspicious activity through email or through an interface. However, these tools are often not able to retain the specific settings required to redo the same or the modified forms of the routine administrative tasks. Specially, these tools do not have a single, unified user interface to manage the entire IT administrative works, including controlling the development, QA and Production environments.

[1] A box is often referred to as a host, server, or in general a system.

Therefore, a tool that would automate and thus speed up such repetitive tasks in a timely manner can increase productivity. However such a tool must be user friendly and flexible enough to be customized per user specifications.

1.1 Birth of the Idea and its Incentives

The main aim of this project is to create a tool that enables remote administration of servers on multiple, disparately located boxes. The tool developed is called Single Interface Remote Administration Tool (SIRAT). The proposed tool will enable an administrator to login to the administrator tool via a single, unified GUI. The administrator will be able to perform a wide range of tasks such as routine maintenance and start/stop services on any or multiple remote computers. The administrator will no longer need to log into several boxes separately to do the routine maintenance work. Also the tool can be customisable and scalable to fit each administrator's unique requirements. Such a utility with a common GUI interface to connect to any or all the servers remotely, to do most of the common administrative or daily routine tasks would provide immense value to any IT firm. This in turn will save time and money.

There are not many tools in the market to meet these demands. During research we could see that this utility can be built by using third party remote administration tools available in the market. Such tasks as described above can be easily accomplished through the use of scripts; however the script language comes with its own pitfalls. The primary disadvantage of scripts is the lack of a user friendly GUI (Graphical User Interface). Security can also be a concern with scripting language; while using scripts one has to openly mention the passwords in a file. These concerns can be solved by designing a user friendly tool to help administrators with their daily tasks. This type of tool can be developed to be an interactive tool with a simple yet functional front-end. Through the use of creative messaging and interfaces, a user can view the activities that are taking place after a task is initiated. It will provide higher levels of security as the passwords will not be openly exposed in a text file but will be stored in a database and not understandable or visible to the naked eye.

The boundary of the topic is very vast, so for the scope of this project we will be limiting this idea to the middleware world only. There are two major technologies that occupy **most** of the Middleware world. One is Dot Net (.Net) and the other is Java. Dot net applications can run as services and daemons depending upon the operating system used. On the other hand, Java world runs on Java virtual machine (JVM). JVM's are bundled with applications servers. Some of the most popular application servers are IBM Websphere, Oracle Weblogic, JBoss, and Tomcat. In this project, we will consider one of these technologies to showcase how the tool will be used in doing the daily

administrative tasks of the application server. Additional extra features can be looked into, as the project develops.

2 SIRAT Functional Specifications

2.1 Functions Performed

As the scope of the topic is very vast, this project will be limited to windows operating system only. In windows operating system, everything can be done using a batch file or as a windows service.

2.2.1 Windows Service

On Microsoft Windows operating systems, a *Windows service* is a long-running executable task that performs specific functions, which is designed not to require user intervention. Windows services can be configured to start when the operating system is booted and run in the background as long as Windows is running, or they can be started manually when required. They are similar in concept to a UNIX daemon. Many of such services appear in the processes list in the *Windows Task Manager*, most often with a username of *SYSTEM*, *LOCAL SERVICE* or *NETWORK SERVICE*, though not all processes with the SYSTEM username are services. The remaining services run through *svchost.exe* as *Dynamic Link Libraries* (DLLs) loaded into memory. Once a service is installed, it can be managed by launching "Services" from the *Windows Control Panel* → *Administrative Tools* or typing "Services.msc" in the *Run* command on Start menu. In Windows Vista and later, it can be started or stopped in the Services tab in Windows Task Manager, and its process can also be found. The "Services" management console provides a brief description of the service functions and displays the path to the service executable, its current status, startup type, dependencies and the account under which the service is running

2.2.2 Windows Batch Files

In DOS, OS/2, and Microsoft Windows, a batch file is a text file containing a series of commands intended to be executed by the command interpreter. Similar to job control language and other systems on mainframe and minicomputer systems, batch files were added to ease the work required for certain regular tasks by allowing the user to set up a batch script to automate many commands. When a batch file is run, the shell program (usually *command.com* or *cmd.exe*) reads the file and executes its commands, normally line-by-line. Batch files are useful for running a sequence of executables automatically and are often used to automate repetitive or tedious processes. Unix-like operating systems (such as Linux) have a similar type of file called a shell script. DOS batch files

have the filename extension *.bat*. Batch files for other environments may have different extensions, e.g. *.cmd* or *.bat* in the Microsoft Windows NT-family of operating systems and OS/2, or *.btm* in 4DOS and 4NT related shells. The Windows 9x family of operating systems only recognizes the *.bat* extension. In OS/2, a file with a *.cmd* extension can also be a *Rexx* file. In MS-DOS, a batch file can be started from the command line by typing its name (along with any required parameters) and pressing the "enter" key. When MS-DOS loads, the file *AUTOEXEC.BAT* is automatically executed, so any commands that need to be run to set up the MS-DOS environment could be placed in this file. Computer users would have the autoexec file set up the system date and time, initialize the MS-DOS environment, load any resident programs or device drivers, or initialize network connections and assignments.

So a remote administrative tool should be able to handle both Windows batch files and Windows Services on the remote boxes. For example if you have a MySQL database server running on a remote machine as Windows Service, then the administrator through SIRAT should be able to Start or Stop or Query the Windows service remotely. In the same way if a Websphere Server can be started and stopped by using a batch file then SIRAT should be able to execute the batch file and should be able to get the status of the commands executed in the batch file.

2.2 User Interface Design

The goal of user interface design is to make the user's interaction as simple and efficient as possible and accomplish user goals - this is often called user-centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilized to support its usability. The design process must balance technical functionality and visual elements to create a system that is not only operational but also usable and adaptable to changing user needs. To enable restart, and thus reuse of the same services, the interface should be able to collect and store the data in a database. Furthermore, a secure authentication system is needed to limit access to the trusted few personnel only. SIRAT attempts to achieve these demands.

2.3 Integration with other Technologies

The design and development of SIRAT involves its integration with several technologies. One such technology is Oracle Weblogic. It will be demonstrated how SIRAT can be used in administrating Weblogic. In a real multinational IT firm, there are several Weblogic Domains to control. In each domain one might have several servers where the applications are deployed. It is assumed that the reader is familiar with rudimentary Weblogic administration. Some batch scripts need to be customized to observe the status of the script’s output. This customization and other details will be explained in the design section.

2.4 Error Conditions and System Messages

Every tool or system developed should be in a position to handle exceptions. Not only these tools need to handle exceptions but also a good user friendly tool should give a proper explanation of the error on the interface. It is essential for the administrator to be aware of the status of the task executed, i.e. whether the task executed successfully or failed. This monitoring capability is of extreme importance as in real world situations, such tools will be used in production environments. The error handling capability just described is a critical feature of SIRAT.

2.5 Additional Functionalities

A tool such as SIRAT will be of great benefit if some of the most commonly used administrative tasks are embedded in the tool. Within this perspective, the following functionalities come to mind:

1. Ping a remote Machine
2. Obtain the configuration of a remote machine
3. Gather memory related information of a remote machine
4. Find disk usage of a remote machine
5. Observe the Task Manager of a remote machine

These functionalities will prove useful for troubleshooting. For example, SIRAT can be used to determine whether the reason behind the failure of a batch file execution is due to lack of sufficient memory storage or lack of disk storage.

2.6 Help Material and User Manual

As a beginner, one should not have to suffer through a lot of guesswork to become familiar with a tool. The portal should have simple and concise instructions to make using the tool a layman's job! Users should be able to understand the functionalities and navigation through the tool. The user manual should cover all aspects about the tool. At the same time, the material should be short and precise so that less time need be spent on reading and assimilating it. Such needed documentation is kept in mind while developing SIRAT.

2.7 Limitations and Restrictions

This section is written to outline some of the features/functionalities of SIRAT that are not covered in this project. SIRAT can be developed to be multifunctional. It can span

various topics and functionalities, covering all aspects of administrative needs. This can be infeasible within the time allocated for this project. Therefore, SIRAT has been created as a plug-in or add-on tasks. However, SIRAT can be customized to include enhanced functionalities and be integrated with other technologies.

1. **Operating System** – SIRAT is mainly developed for Windows operating System. The third party tools used will only work with windows OS.

2. **Authorization** – SIRAT is limited to authentication facility only. It does not have the facility for authorization. For the purpose of this project, the focus has been on developing the tool's framework and main functionalities rather than concentrating on other probable enhancements.

3. **Look and Feel** – Overall the look and feel of SIRAT can be improved. Currently the tool has met the standard design specifications.

4. **Password Encryption** – All the passwords entered in the portal are stored in the database tables. The stored passwords in the database are in plain text format. They are not encrypted.

3 SIRAT Performance Requirements

3.1 Efficiency

"Efficiency" might have varying meanings in different disciplines. Efficiency in general describes the extent to which time or effort is well used for the intended task or purpose. It is often used with the specific purpose of relaying the capability of a specific application of effort to produce a specific outcome effectively with a minimum amount or quantity of waste, expense, or unnecessary effort. Proper care should be taken while building a tool to have its efficiency at an acceptable level.

When it comes to SIRAT, the efficiency is measured with respect to the end user. How the end user can do his daily administrative tasks with and without SIRAT. We expect that the administrator's efficiency will be improved when he or she uses SIRAT.

3.2 Description of Reliability Measures

Proper care has been taken while building SIRAT for reliability. When a Windows service is started on a remote machine, SIRAT has an option to query the status of the service. For example when a command is executed to start a Windows Service on a remote machine, SIRAT has the option to run a query command to find out the status of the service executed. In this manner, it can be determined whether the service executed

successfully. For a Windows Batch File, the result of the batch file can be routed to a log file so that the status of each command in the batch file can be checked by reading the log file.

3.3 Error/Failure Detection

SIRAT has a good capability of handling errors and failures. It is assumed that the errors and failures are not from the software's used to build this tool. If there is any failure while using SIRAT, the error log is output in the portal with the self-explanatory error messages or exception messages. Even the tool logs can be used for more rigorous investigation.

3.4 Security

SIRAT's user group is very limited to a few users (administrators) who are trustworthy to use the tool. SIRAT has a simple multilevel security in place. Other than the authentication through userid/password to get access to SIRAT, the tool is designed not to allow access to remote boxes for activities that might be considered a security breach, unless the user is given the userid/password of the remote box. There are other security measures that are left for future implementation. For instance, one way to minimize malicious behavior or discourage the user from malicious activities is to log the user activities to a separate file to which the user has no access. That way, the file can be analyzed for suspicious behavior or can be used for debugging in cases of failure.

3.5 Maintainability

SIRAT can be easily maintained with minimum technical skills. It does not use many resources to make it work. Once installed and configured, the minimum knowledge needed is to learn how to start/stop JBoss and MySQL servers. Furthermore, SIRAT can be installed and configured on a normal laptop or desktop without many resources from the host.

4 SIRAT Design Overview

This section will explain about SIRAT architecture and how the third party software packages are integrated in building it.

4.1 SIRAT Design Architecture

SIRAT is a java based application using third party utilities to communicate with different remote machines. The SIRAT portal is hosted on JBoss application server. It uses JRockit

as its JVM. The back-end database is MySQL. It uses PsTools suite to communicate between remote machines. Figure 1 shows the high level architectural diagram of SIRAT.

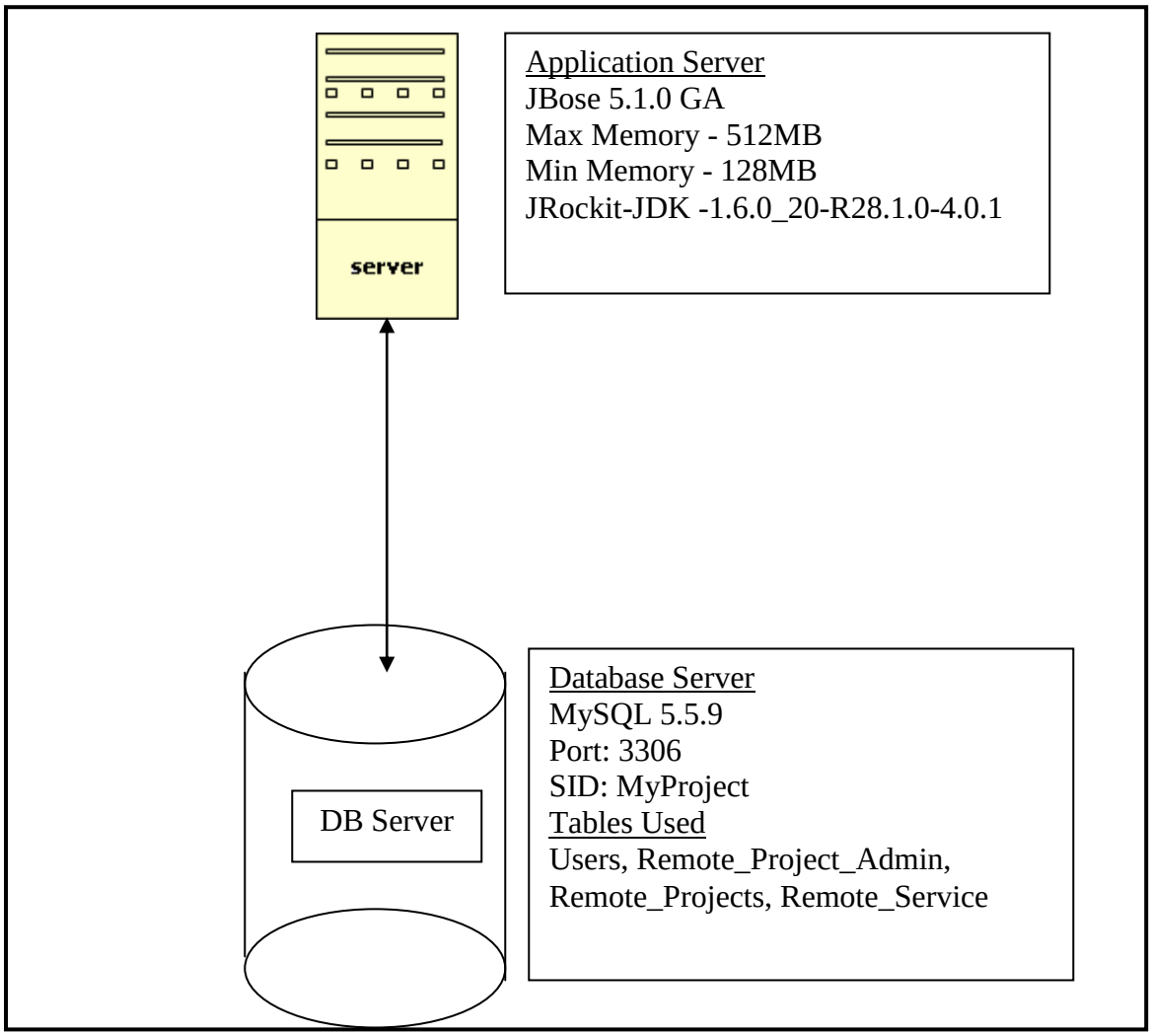


Figure 1 SIRAT Architectural Diagram

The PsTools suite includes several individual applications, which can be run on command line to get or to do various administrative tasks on the local or the remote machines. In order to make PsTools Suite work, the below are pre conditions which should be implemented on the network you are running SIRAT;

1. **Ping** – The Remote machine should give positive response when you ping it. That means all the Remote Machines should in a reachable network
2. **Net share** - ADMIN$ share should be enabled. Means all the Remote Machine's C:\WINDOWS folder should be accessible from your work station.

3. **Remote Registry service** - PsTools suite makes use of RPC calls via port 445. Make sure that the port 445 is opened in the firewall.

4. **Administrator Rights** – Make sure that you have Administrator rights on all the Remote Machines you are accessing.

There is more written about PsTools suite in the appendix section. Figure 2 explains the communication process of PsTools Suite.

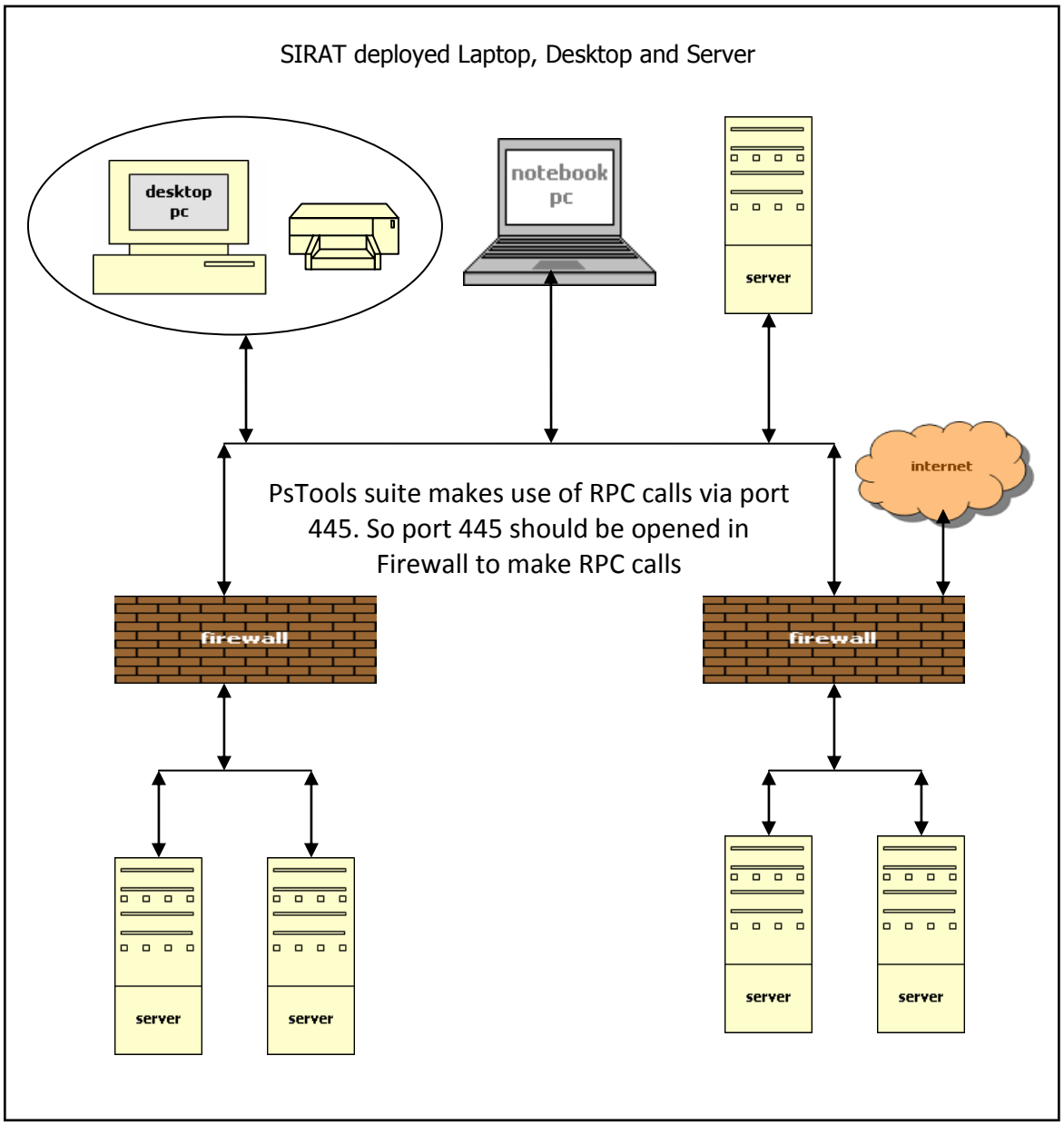


Figure 2 SIRAT Communication Channel

SIRAT can be used as a single-user deployed in a personnel laptop/desktop or it can be deployed on a server by hosting the URL to different users. It is not recommended to host SIRAT for multiple users as authorization is not enabled and session problems might arise.

4.2 SIRAT User Interface Terminology

SIRAT user interface is made as simple as possible. In the context of SIRAT, the term *service* refers to do an activity on the remote server from SIRAT. The activity can be anything like starting or stopping a database or an application server on the remote machine. As mentioned earlier SIRAT is developed for Windows Operating system. So a service can be a start, stop, query a Windows Service, or execute a batch file on the remote machine. All the services in SIRAT will be branched under a *project.*

The term *project* refers to a group of services with similar or unique characteristics. This can be best explained by quoting an example. In SIRAT you create a project called Middleware or DBA. Create all the services which belong to middleware category under project Middleware. At the same time all the services which come under Database Administration will come under the project DBA.
You can create a project with whatever name you want but it should make sense to the services under it. The main purpose is to categorize all the services in a manageable form.

4.3 SIRAT Data Flow Diagram

The flowchart in Figure 3 describes the current SIRAT user interface usage.

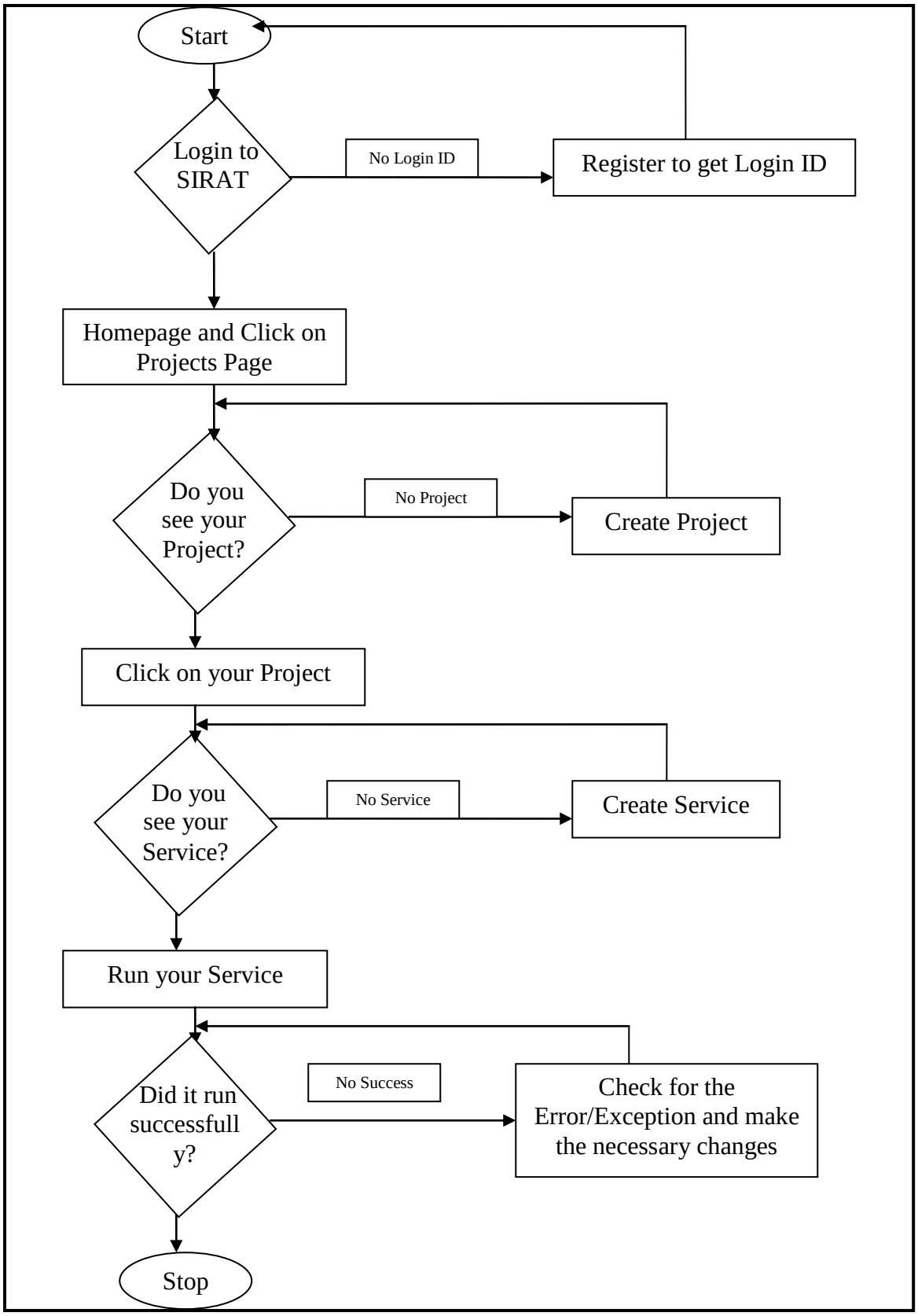


Figure 3 SIRAT Flow Chart Diagram

4.4 SIRAT Database Tables

SIRAT has mainly four database tables. The four tables are listed below;

- Users
- Remote_Projects
- Remote_Service
- Remote_Project_Admin

Users table is created to store all the user information of SIRAT. Apart from user information it also stores the user credentials.

Remote_Projects is created to store all the project names and its details.

Remote_Service is used to store the information while creating a service. The information can be later used to run the service.

Remote_Project_Admin table stores the information needed while navigating through SIRAT. It is also used while running a service.

5 SIRAT Installation and Configuration Overview

5.1 Software Packages used in building SIRAT

SIRAT is built using software packages with freeware license. As indicated, the communication among remote machines is done by PsTools suite. The web-portal is developed on Eclipse and deployed on JBoss version 5.1.0 GA. The database is created and managed on MySQL 5.5.9.

The list of software packages utilized to build SIRAT is given below:

1. **MySQL 5.5.9**
2. **JRockit 1.6.0_20**
3. **JBoss 5.1.0 GA**
4. **PsTools Suite**

All these technologies are well discussed in the appendix section.

SIRAT can be installed on a user desktop/laptop or on a server. A firm can decide to deploy SIRAT on a server and host the URL to all its users, thereby enabling the administrator of the firm to see all the several services on a single portal. This scenario is well suited for small firms with limited number of services and with few people handling most of the administrative activities. Hosting is not recommended at this point as

authorization and session handling is not enabled in this version of SIRAT. SIRAT deployment does not require highly configured laptops or servers. It uses minimum resources so it can be deployed on any machines with minimal configuration.

5.2 Installations

The detailed installations of the packages are described in the appendix section. So in Section 5.3, it is assumed that the installations of these packages are complete.

5.3 Configuration

5.3.1 Configure MySQL 5.5.9

Before using SIRAT, the database needs to be configured in the following manner:

1. Start the MySQL windows service. This will start the MySQL Database.
2. Connect to the database by "*mysql –u user -p*". Here the user is 'root' and password is 'MyProject'.
3. Create a database called 'myproject'. Create the database by using the command "*CREATE DATABASE myproject;*" As MySQL can contain multiple databases, to use the myproject database, after creating the database, the following command must be followed: "*USE myproject*". Now you can start using the database "myproject".
4. Create the tables for myproject required for SIRAT. SIRAT mainly requires four tables. The Data Dictionary on creating the four tables are given below;

5.3.1.1 Table Creation

- **Remote_Project_Admin** - This table is used to store the information which is required while executing a service. Most of the attributes used here are self explanatory.

```
CREATE TABLE myproject.remote_project_admin(
   id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
   PROJ_ID varchar(30),
   SERVICENAME varchar(30),
   BOX_NAME varchar(30),
   DISCRIPTION varchar(100),
   USERNAME varchar(30),
   USER_PASSWORD varchar(30),
   OS varchar(30),
   REQUEST_TYPE varchar(30),
```

```
    SERVICE_TYPE_NAME varchar(30)
)
```

- *ID and PROJ_ID* are used internally by the java code to maintain the session.
- *SERVICENAME* is the name given to the service by the user.
- *BOX_NAME* is the DNS name or the IP address if the remote Machine.
- *DISCRIPTION* is used to describe what the service does.
- *USERNAME* and *PASSWORD* is used to store the username and password of the remote machine.
- *OS* is the operating system of the remote machine.
- *REQUEST_TYPE* is used to know if the request is to run a windows service or to run a batch file on the remote machine.
- *SERVICE_TYPE_NAME* – if Request_Type is batch it stores the full path of the batch file and if Request_Type is Windows service than it store the name of the Windows service on the remote machine.

- **Remote_Service** - This table will store the information when the user inputs the service information data for the first time in the Create Service Page.

```
CREATE TABLE myproject.remote_service(
   id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
   PROJ_ID varchar(32),
   SERVICENAME varchar(128),
   BOX_NAME varchar(30),
   DISCRIPTION varchar(300),
   USERNAME varchar(64),
   USER_PASSWORD varchar(64),
   OS varchar(32),
   REQUEST_TYPE varchar(32),
   SERVICE_TYPE_NAME varchar(300)
)
```

- *ID and PROJ_ID* are used internally by the java code to maintain the session.
- *SERVICENAME* is the name given to the service by the user.
- *BOX_NAME* is the DNS name or the IP address if the remote Machine.
- *DISCRIPTION* is used to describe what the service does.
- *USERNAME* and *PASSWORD* is used to store the username and password of the remote machine.
- *OS* is the operating system of the remote machine.
- *REQUEST_TYPE* is used to know if the request is to run a windows service or to run a batch file on the remote machine.
- *SERVICE_TYPE_NAME* – if Request_Type is “batch”, it stores the full path of the batch file and if Request_Type is “Windows service”, then it stores the name of the Windows service on the remote machine.

- **Remote_Projects** – This table is used to store the Projects created information.

```
CREATE TABLE myproject.remote_projects(
   PROJ_ID INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
   PROJ_NAME varchar(128),
   PROJ_DISC varchar(300)
)
```

  - *PROJ_ID* – For each project created in SIRAT, an ID is associated with it. This ID will be used by the Java Code.
  - *PROJ_NAME* is used to store the name of the project created in SIRAT.
  - *PROJ_DISC* is used to store the description about the project.

- **Users** – This table is created to store all the user information. Apart from user information it also stores the user credentials.

```
CREATE TABLE myproject.users(
   USER_ID INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
   USER_NAME varchar(32),
   USER_PASSWORD varchar(64),
   FIRST_NAME varchar(64),
   LAST_NAME varchar(64)
)
```

  - *USER_ID, USER_NAME*, *USER_PASSWORD, FIRST_NAME* and *LAST_NAME* – As the variables used here are self explanatory. All are related to user information.

5. Once the four tables are created, the configuration of the Database is complete

Note: In JBoss configuration section you should be ready with the drivers used to make connection from the application server to the database.

5.3.2 Configure JBoss Application Server

This section will describe the JBoss configuration required for deploying SIRAT. By this time we should be ready with the final java code (RemoteAdmin.war). Follow the instructions for deploying SIRAT:

1. Copy the JDBC driver required for the JBoss datasource. Download MySQL driver for JBoss version you are using and copy it in C:\RemoteAdmin\jboss-5.1.0.GA\server\default\lib folder. The name of the file will be "mysql-connector-java-5.1.10.jar".

*Note: A **JDBC driver** is a software component enabling a Java application to interact with a database. To connect with individual databases, JDBC (the Java Database Connectivity API) requires drivers for each database.*

2. Now copy the DataSource xml file which has required information of the database. The name of the file given here is "rpa-jboss-ds.xml". Copy this in "C:\RemoteAdmin\jboss-5.1.0.GA\server\default\deploy" folder.

Note: rpa-jboss-ds.xml file is used by the JBoss application server to get the information about the database which it needs to connect.

The content of the file is;

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>jdbc/rpads</jndi-name>
    <connection-
url>jdbc:mysql://localhost:3306/myproject?autoReconnect=true&amp;autoReconnectFo
rPools=true</connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>root</user-name>
    <password>MyProject</password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <idle-timeout-minutes>30</idle-timeout-minutes>
  </local-tx-datasource>
</datasources>
```

3. Create a folder called RemoteAdmin.war in "C:\RemoteAdmin\jboss-5.1.0.GA\server\default\deploy" folder. The name of the folder can be anything you want but it should always have the extension ".war". Now unzip the contents from the RemoteAdmin.war in this folder. Please see the screenshot below with the contents in the folder.

Note: deploy folder is the place where you keep all the applications needed to be deployed on the JBoss server. Figure 4 shows all the folders in the application folder (RemoteAdmin.war) which is placed under deploy folder of JBoss.

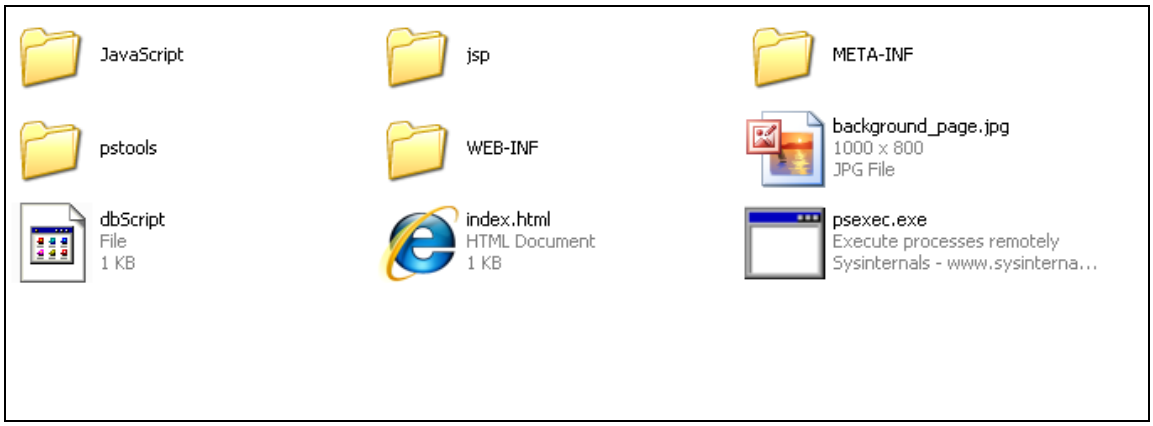


Figure 4 RemoteAdmin.war Folder in the Deploy Folder of JBoss

4. Now in the descriptor file we need to state the place where the PSTools Suite is placed. Go to: "C:\RemoteAdmin\jboss-5.1.0.GA\server\default\deploy\RemoteAdmin.war\WEB-INF" folder and open the "web.xml" descriptor. In the file you need to update as shown in below figure.

Note: web.xml file contains the default settings for all web applications deployed in JBoss server. This file needs some to be modified if you want to incorporate something the application needs. This file will be used to let the application know where the PsTools Suite is installed.

```
<? xml version="1.0" encoding="UTF-8"?>
<web-ap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>RemoteAdmin</display-name>
   <servlet>
        <servlet-name>RemoteProjAdminBusinessBean</servlet-name>
        <servlet-class>com.remoteadmin.businessbean.RemoteProjAdminBusinessBean</servlet-class>
        <load-on-startup>0</load-on-startup>
   </servlet>
   <context-param>
        <description>Path of window execute files.</description>
        <param-name>exe_folder</param-name>
        <param-value>C:\RemoteAdmin\Tools\PSTools</param-value>
   </context-param>
   <welcome-file-list>
        <welcome-file>Login.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

5. Make sure that the PSTools are placed in the folder mentioned in the “web.xml” file. Also run and register all the exe files used inside the folder. Double click the “.exe” files and it should automatically register for you. Figure 5 shows all the executable tools of PsTools Suite.

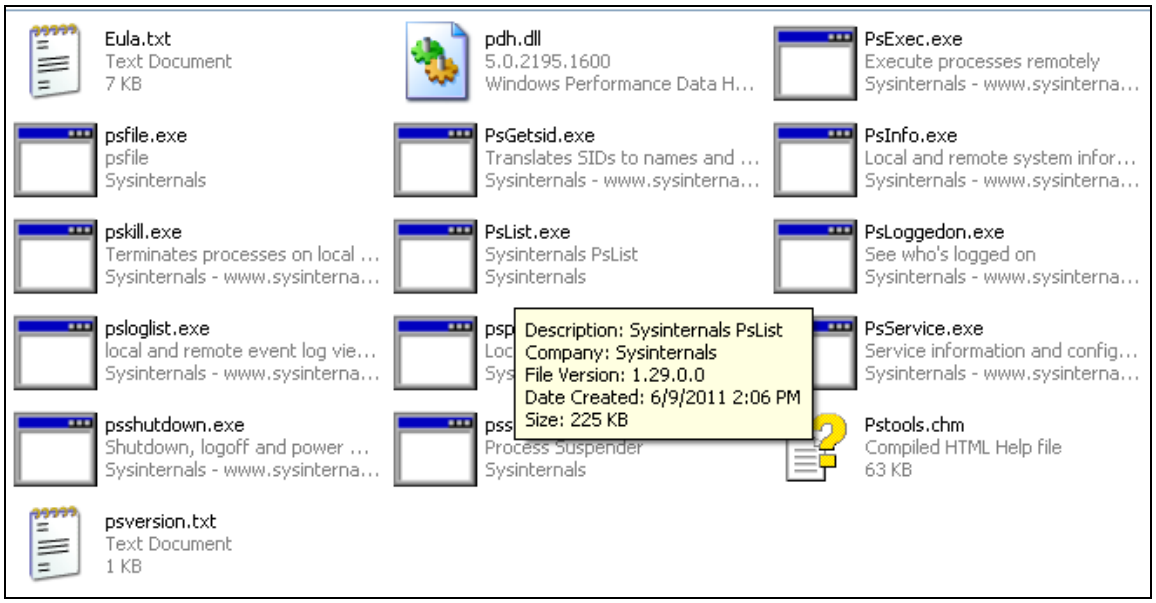


Figure 5 PsTools Suite

6. Now the application server is ready to be started. In order to run the application server, go to “C:\RemoteAdmin\jboss-5.1.0.GA\bin” folder and double click run.bat file. In order to open the login page of SIRAT, type the URL “http://localhost:8080/RemoteAdmin/jsp”.

Note: The bin folder is the place where all the configuration files of JBoss application server can be seen.

5.3.3 Windows Batch File Customization

In order to run a Windows Batch file on a remote machine, some customization has to be done on the file. The steps below show how that is to be done;

1. Create a wrapper batch file to the main batch file which should always end with .cmd extension.
2. The wrapper script should be in the below format:

start /min C:\<fullpath>\myfile.bat

Note: where myfile.bat is the actual batch file we need to start on the remote machine. The option min (minimized) will start the wrapper batch file in minimized mode.

3. On all the remote machines (where SIRAT is used), create a shared folder in C: drive named "*SharedFolder"*. This folder should be shared to all the users who have administrator's rights on the remote machine and with the User ID used to login to the laptop/desktop where the SIRAT is installed. This folder is used to output the batch file logs.
4. The output statements should be logged in the main batch file. That is, it should be in "myfile.bat" file.
5. The output file should always have the same name as the actual file. In this case it should be "myfile.log".

5.3.3.1 Example Batch files

Myfile.bat – This file is used to find the IP Address of the remote machine.

```
@ECHO OFF
@REM Name of the file – Myfile.bat
@REM This script is used to find the IP Address of the remote machine
call C:\Windows\system32\ipconfig.exe > C:\SharedFolder\IPAddress.log
 exit
```

IPAddress.cmd – This is the wrapper script that we need to create to call the main batch file (Myfile.bat) in minimized mode.

```
@ECHO OFF
@REM Name of the file – IPAddess.cmd
@REM This script is used to find the IP Address of the remote machine
start /min <fullpath>\Myfile.bat
exit
```

This completes the installation and configuration of SIRAT. Now the next chapter describes about the user interface of SIRAT.

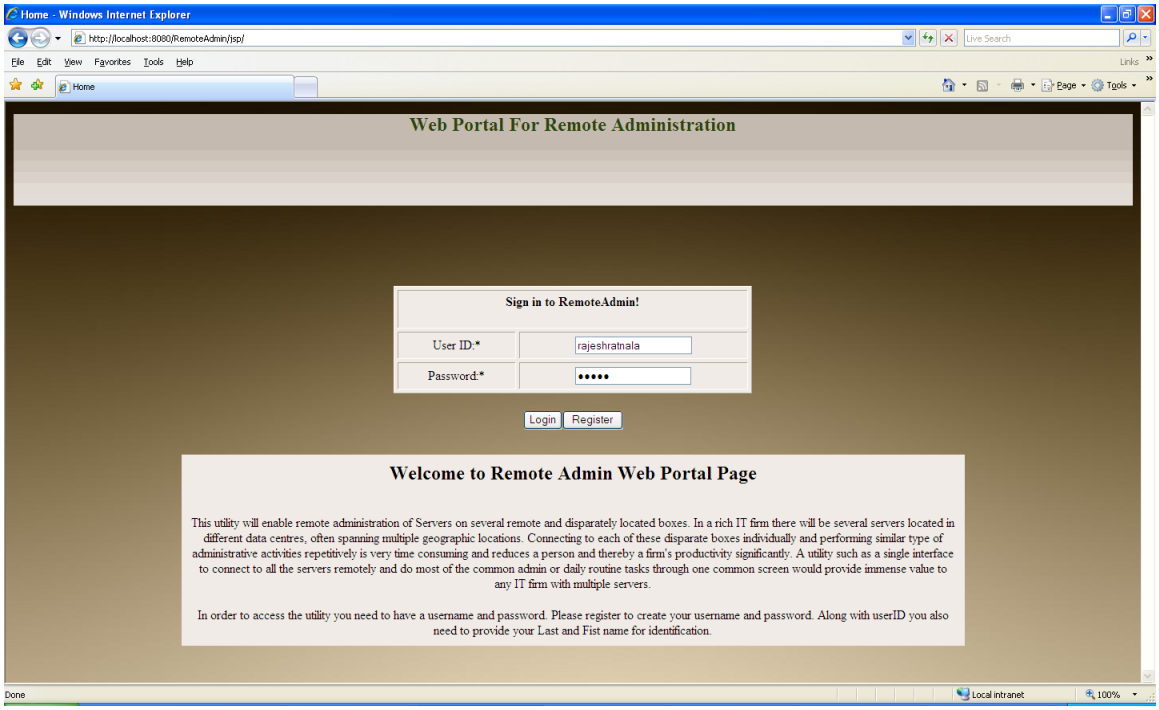
6 SIRAT User Interface Overview

This section will discuss SIRAT's user interface and will use screenshots and an explanation of the various functionalities. When SIRAT starts, the portal shows six buttons at the top for most activities: Home, Projects Created, Create Service, Reachability Test, About, and Logout. . When a user clicks on "Home", it will take the user to the Home Page. Home page contains some description about the tool and the user interface. The second button is "Projects Created", when one clicks on it; it will take one to the page where list of projects created in SIRAT is displayed. In the same page there is an option to create new projects. The third button is "Create Service", when a

user clicks on it, it takes the user to the page where he needs to key in all necessary information to create a service. The Fourth Button is "Reachability Test". When a user selects this button, it will take the user to the page where he can enter information to figure out if the remote box is reachable or get some pertinent information about the performance and health of the remote box. This test can be carried out whenever there are some problems with the remote machine or network. Also during network issues, this page will give confirmed results about the reachability of the Remote Box. The “About” button give some information about the tool and provides the user with documentation. The Logout button is self-explanatory.

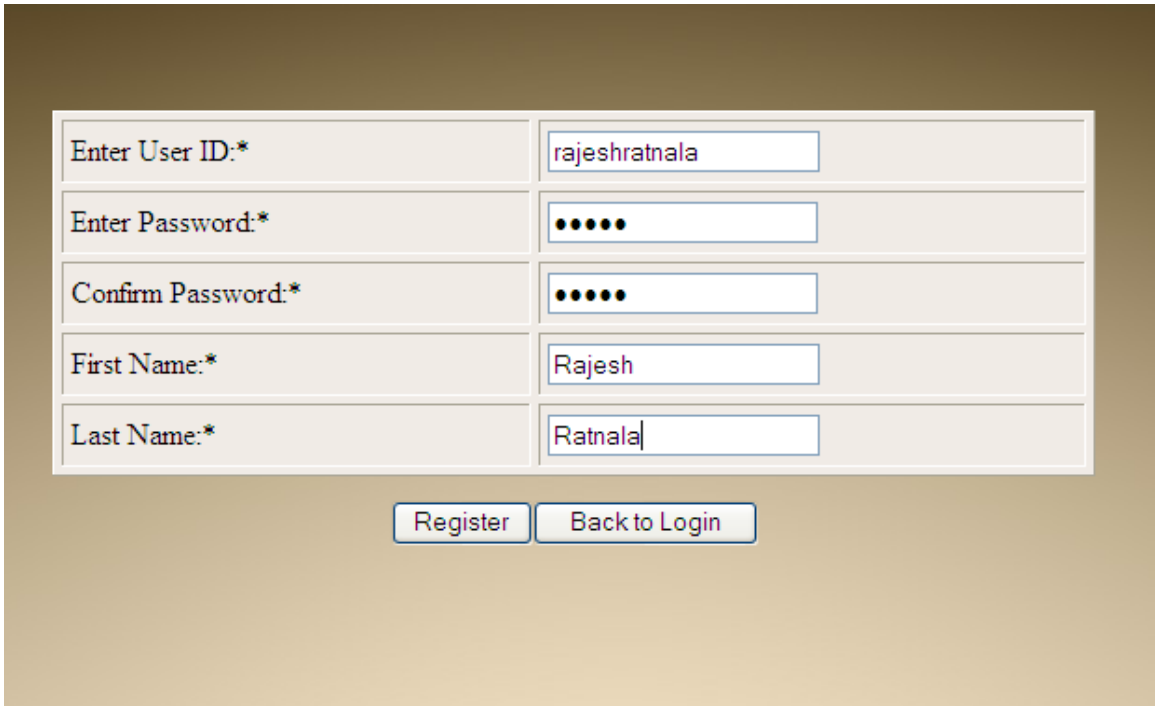
6.1 Login Page

The screenshot below shows the login page of SIRAT. If the user doesn’t have the login credentials, there is a button called “Register” allowing the user to register and get the login credentials.


Home - Windows Internet Explorer
http://localhost:8080/RemoteAdmin/jsp/
File Edit View Favorites Tools Help
Home
Web Portal For Remote Administration
Sign in to RemoteAdmin!
User ID:*
rajeshratnala
Password:*
Login
Register
Welcome to Remote Admin Web Portal Page
This utility will enable remote administration of Servers on several remote and disparately located boxes. In a rich IT firm there will be several servers located in different data centres, often spanning multiple geographic locations. Connecting to each of these disparate boxes individually and performing similar type of administrative activities repetitively is very time consuming and reduces a person and thereby a firm's productivity significantly. A utility such as a single interface to connect to all the servers remotely and do most of the common admin or daily routine tasks through one common screen would provide immense value to any IT firm with multiple servers.
In order to access the utility you need to have a username and password. Please register to create your username and password. Along with userID you also need to provide your Last and Fist name for identification.
Done
Local intranet
100%

6.2 Register Page

This page is used for user registration for SIRAT. User needs to provide the desired login ID, password, confirm password, first name and last name.
Once registration is done the user can go back to Login Page by clicking on the “Back to Login” button.



6.3 Home Page

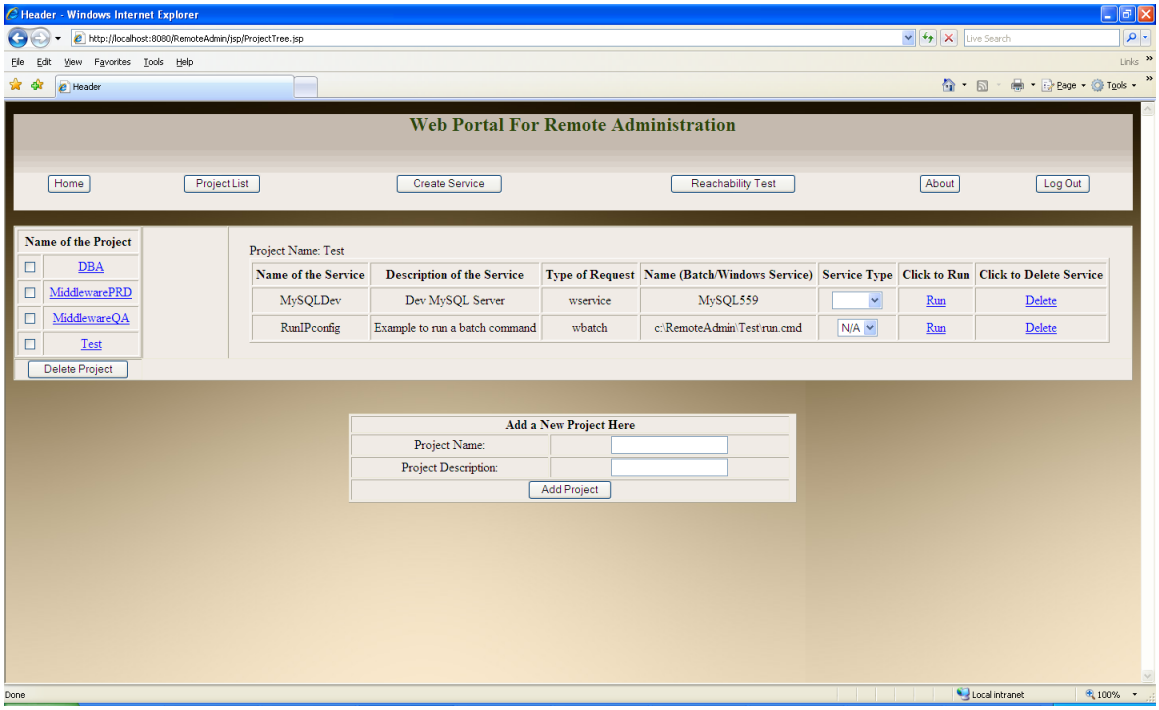
The screenshot below shows the Home Page. This page mainly has content describing SIRAT's usage and the user interface.

Web Portal For Remote Administration
Home
Project List
Create Service
Reachability Test
About
Log Out
Single Interface Remote Administration for Middleware Technologies
This page will give you the basic information about this Remote Web Portal
This utility will enable remote administration of Middleware Servers on several remote and disparately located boxes. In a rich IT firm there will be several servers located in different data centres, often spanning multiple geographic locations. Connecting to each of these disparate boxes individually and performing similar type of administrative activities repetitively is very time consuming and reduces a person and thereby a firm's productivity significantly. A utility such as a single interface to connect to all the middleware servers remotely and do most of the common admin or daily routine tasks through one common screen would provide immense value to any IT firm with multiple servers.
There are mainly five Tabs at the top of every page in this Portal. When you click on "Home", it will always bring you to the current page you are in right now. The second button that is "Projects Created", when you click on it, it will take you to the page where it will show you all the list of the Projects created. The third button is "Create Project", when you click on it, it will take you to the Page where you are asked all the information to create your soft link. The Fourth Button you can see is "Reachability Test". When you click on it,it will take you to the page where you need to enter information to know if you can reach to the remote box or not. This test has to be done before you create any projects. Also in times of network issues, this page will give confirmed results about the reachability of the Remote Box. The final and the fifth button you see will have all the documention about this tool.

6.4 Project List Page

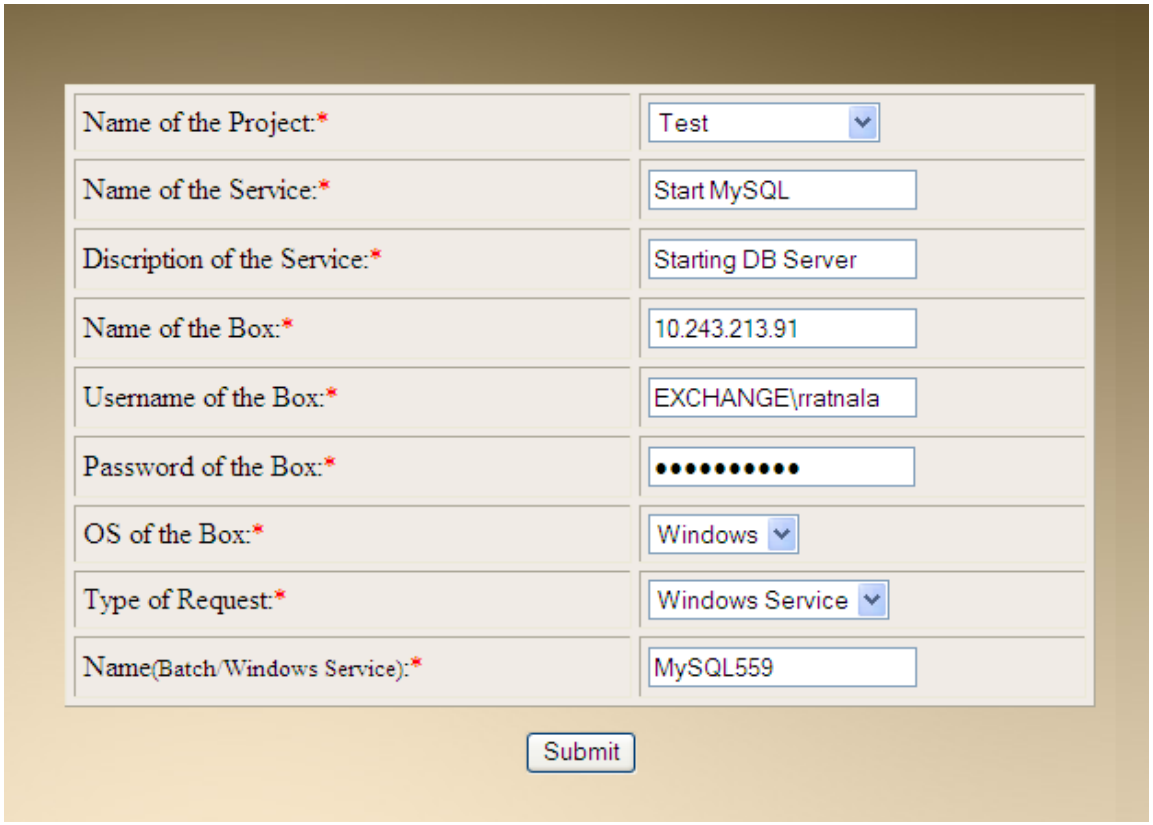
On the left hand side of "Project List" page, there is list of all the projects created in SIRAT. When clicked on each project link, on the right side it will display all the services present for that project. If it is required to switch a project then click on the desired

project name which has the corresponding services. This page also provides the option of allowing a user to create a new project. At the center bottom, provide the project name and project description to create a new project. Once the required information is supplied, click on “Add Project” button. Once this is done, the new project will be added to the list of the projects.



6.5 Create Service Page

This page will allow creating the required services for a listed project. SIRAT’s scope is contained to Windows only. In Windows, using only two types of functions, i.e. a windows service or a batch file, any type of administrative activities can be accomplished. The two screens shots below show the fields that are required to create a service.

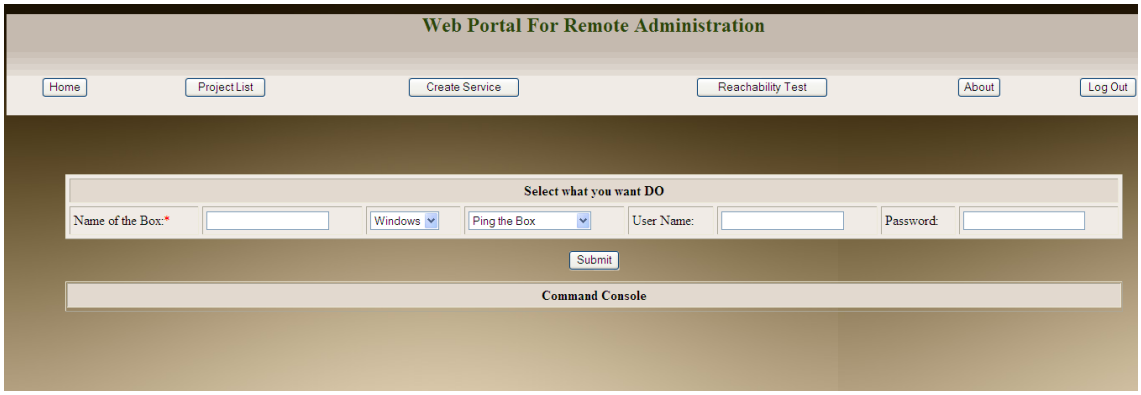


6.6 Reachability Page

This reachability test gives the opportunity to check the resources and the health of the remote box. For example, reachability test can be used to check on the remote machine in case something goes wrong in the middle of executing some services. This also has the advantage to check on the remote machine without directly login into the remote box.

The basic functionalities that are attached to the Reachabilty Page are:

- Ping the Box
- Box Configuration
- Box Memory Usage
- Box Disk Usage
- Box Task Manager



6.7 About and Log Out Pages

The About page will have a document attached, which will have the complete details about SIRAT. This project report will be attached to the About link.

The Log Out button is self-explanatory, clicking on the “Log Out” button will log you out of SIRAT and take you to the Login Page.

7 SIRAT Verification

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

7.1 Test Bed Environment

In order to test SIRAT a network with specification mentioned in the design chapter is needed. As SIRAT uses a third party utility called PSTools Suite, the network should comply with the specifications that are necessary by PSTools Suite. Two networks have been used for the testing, one the office network, where I work, which is very huge, and the second network is a small home network with no more than two remote machines connected to a switch. Testing has been completed on both the networks. The office network is used to show-case that the tool can be used in real world situations. The home network is used to give demonstrations whenever necessary. The two network diagrams are shown below.

7.1.1 Simple Home Network

Figure 6 below shows the design of a simple network at my home. This home-network will also be used during the demonstration at the time of defense of this project.

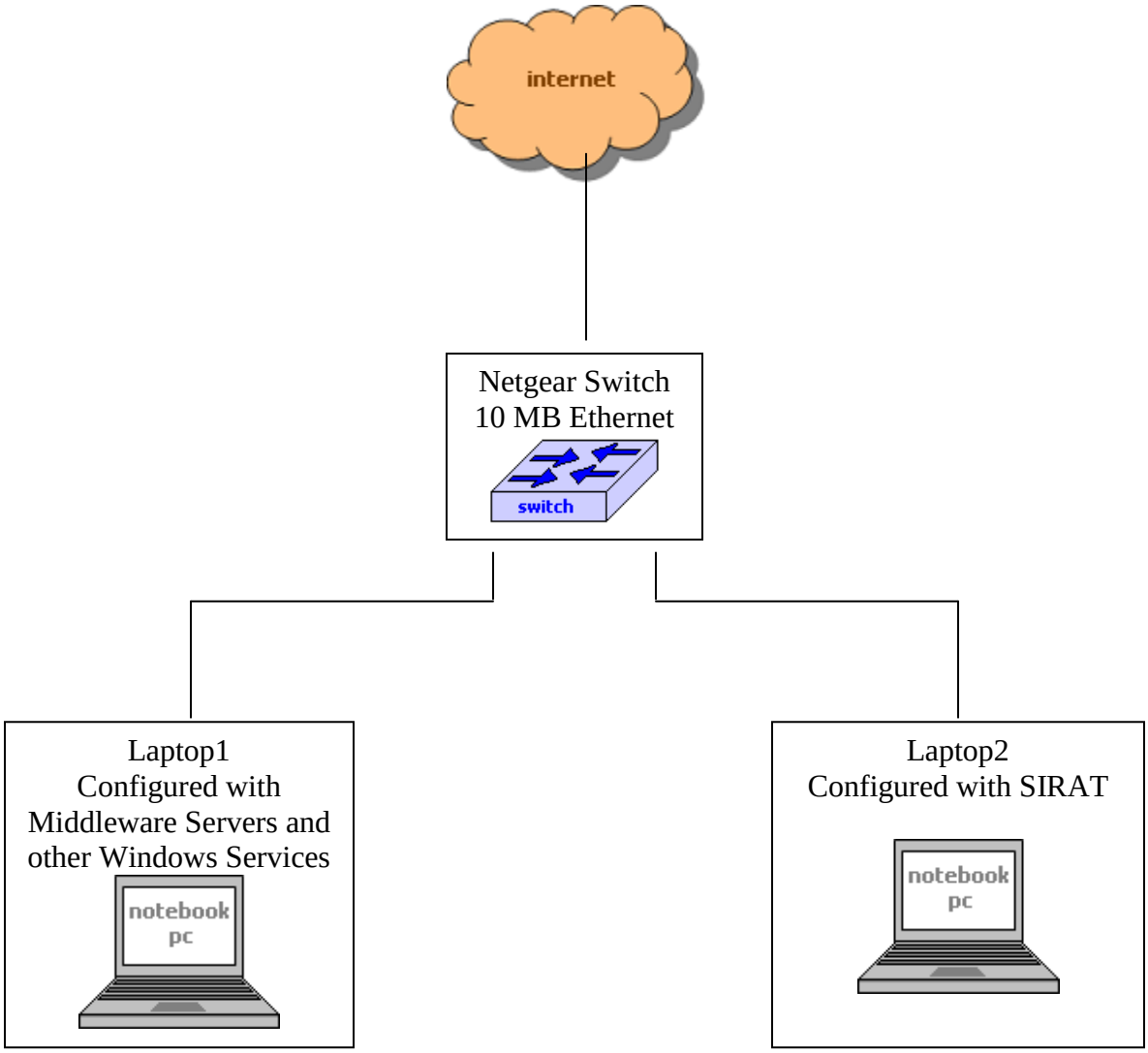


Figure 6 Simple Home-Network for testing SIRAT.

7.1.2 Office Network

The Figure 7 below exhibits the setup in the office-network.

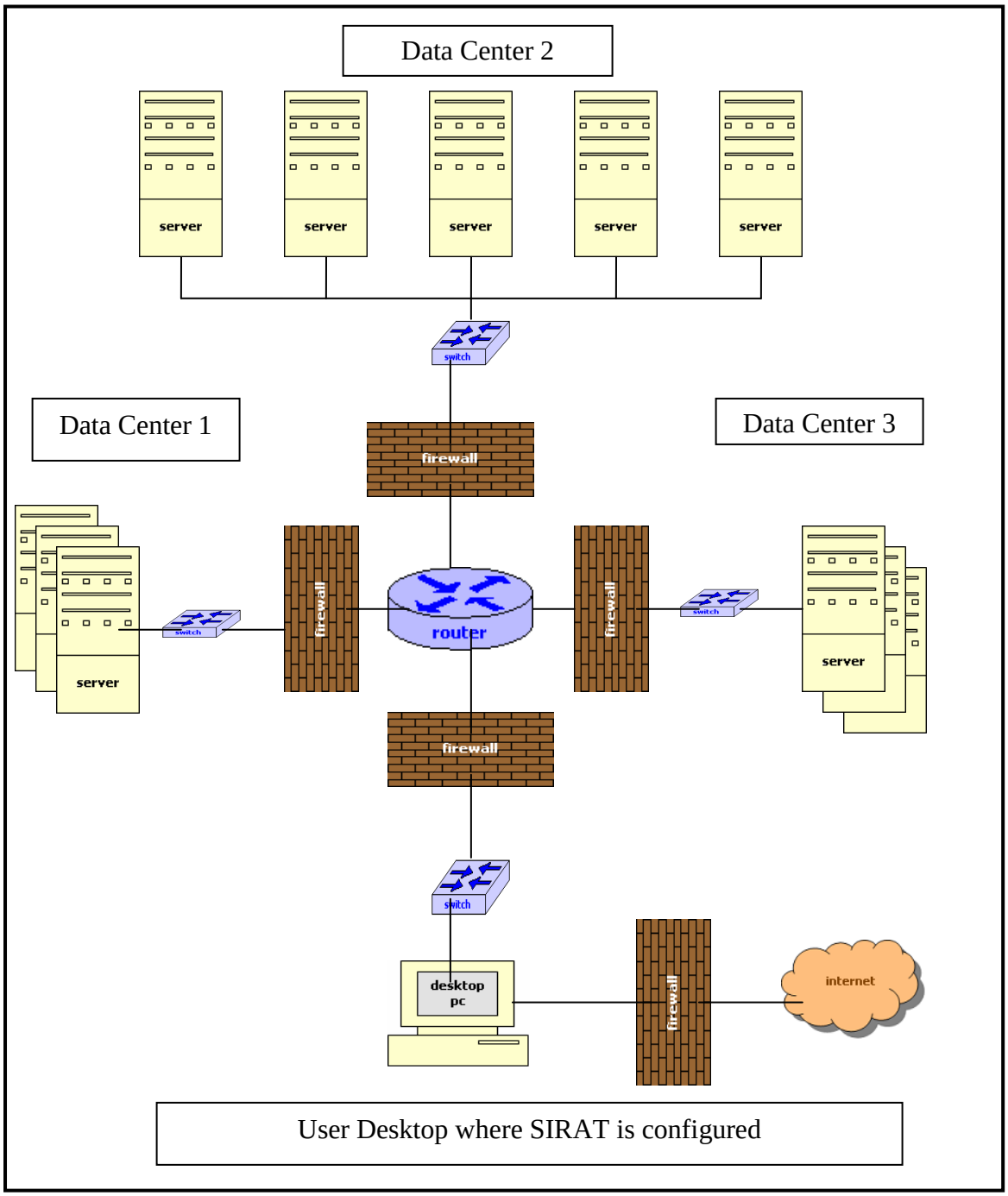


Figure 7 Office Network

7.2 Items/Functions to be Tested

The following is a complete list of items and functionalities to be tested on SIRAT.

1. Registration Process – Create a “testuser” ID
2. Login/Logout to the Tool
3. Create Project
4. Create a Windows Service to a particular Project
5. Create a Batch Service to a particular Project
6. Test the Windows Service created to a particular Project
7. Test the Batch Service created to a particular Project
8. Test Ping the Remote Machine functionality
9. Test Box Configuration of the Remote Machine functionality
10. Test Box Memory Usage of the Remote Machine functionality
11. Test Box Disk Usage of the Remote Machine functionality
12. Test Task Manager of the Remote Machine functionality
13. Test Weblogic Integration with SIRAT

7.3 Description of Test Cases

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test.

The test cases prepared for testing the items 1-13 listed in section 7.2 are:

1. Test Case 1 – Test Registration
    - Open the Interface
    - Click of Register Button
    - Fill in the required data
    - Click Register Button
    - Click Back to Login
2. Test Case 2 – Test Login
    - Open the Interface
    - Enter User ID
    - Enter correct password
    - Click Login Button
3. Test Case 3 – Test with false login credentials

- Open the Interface
- Enter User ID
- Enter incorrect password
- Click Login Button

4. Test Case 4 – Test Create Project
    - Open the Interface
    - Enter login credentials
    - Click Create Project Page
    - Enter new project name and description
    - Click Add Project button
5. Test Case 5 – Test Create Windows Service for a project
    - Open the Interface
    - Enter login credentials
    - Click Create Service Page
    - Enter the necessary information for Windows Service
    - Click Submit button
6. Test Case 6 – Test Create Windows Batch for a project
    - Open the Interface
    - Enter login credentials
    - Click Create Service Page
    - Enter the necessary information for Windows Batch File
    - Click Submit button
7. Test Case 7 - Test Create Windows Service for a project with false information
    - Open the Interface
    - Enter login credentials
    - Click Create Service Page
    - Enter the necessary information for Windows Service with some wrong information
    - Click Submit button
8. Test Case 8 - Test Create Windows Batch for a project with false information
    - Open the Interface
    - Enter login credentials
    - Click Create Service Page
    - Enter the necessary information for Windows Batch File with some wrong information
    - Click Submit button
9. Test case 9 – Test Query Windows Service
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Select Query on Service Type which has correct information

- Click on Run link

10. Test case 10 - Test Start Windows Service
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Select Start on Service Type which has correct information
    - Click on Run link
11. Test case 11 - Test Stop Windows Service
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Select Stop on Service Type which has correct information
    - Click on Run link
12. Test case 12 - Test Query Windows Service with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Select Query on Service Type which has wrong information
    - Click on Run link
13. Test case 13 - Test Start Windows Service with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Select Start on Service Type which has wrong information
    - Click on Run link
14. Test case 14 - Test Stop Windows Service with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Select Stop on Service Type which has wrong information
    - Click on Run link
15. Test case 15 - Test Windows Batch File
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Click on Run link of the Batch file selected which has correct information

16. Test case 16 - Test Windows Batch File with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Project List Page
    - Click on the desired project link
    - Click on Run link of the Batch file selected which has wrong information
17. Test case 17 – Test Ping the Box
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information and select Ping the Box
    - Click on submit button
18. Test case 18 - Test Ping the Box with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information with some wrong input and select Ping the Box
    - Click on submit button
19. Test case 19 - Test Box Configuration
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information and select Box Configuration
    - Click on submit button
20. Test case 20 - Test Box Configuration with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information with some wrong input and select Box Configuration
    - Click on submit button
21. Test case 21 - Test Box Memory Usage
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information and select Box Memory Usage
    - Click on submit button
22. Test case 22 - Test Box Memory Usage with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page

- Enter the required information with some wrong input and select Box Memory Usage
- Click on submit button

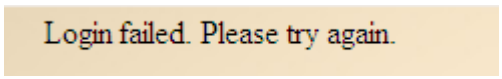
23. Test case 23 - Test Box Disk Usage
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information and select Box Disk Usage
    - Click on submit button
24. Test case 24 - Test Box Disk Usage with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information with some wrong input and select Box Disk Usage
    - Click on submit button
25. Test case 25 - Test Task Manager
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information and select Box Task Manager
    - Click on submit button
26. Test case 26 - Test Task Manager with wrong information
    - Open the Interface
    - Enter login credentials
    - Click Reachability Test Page
    - Enter the required information with some wrong input and select Box Task Manager
    - Click on submit button
27. Test case 27 – Test the integrated Weblogic Application Server Admin Scripts
    - Open the Interface
    - Enter login credentials
    - Create new Project and new Services for integrating with Weblogic Application Server
    - Run the appropriate services to test the utility

7.4 Test Results

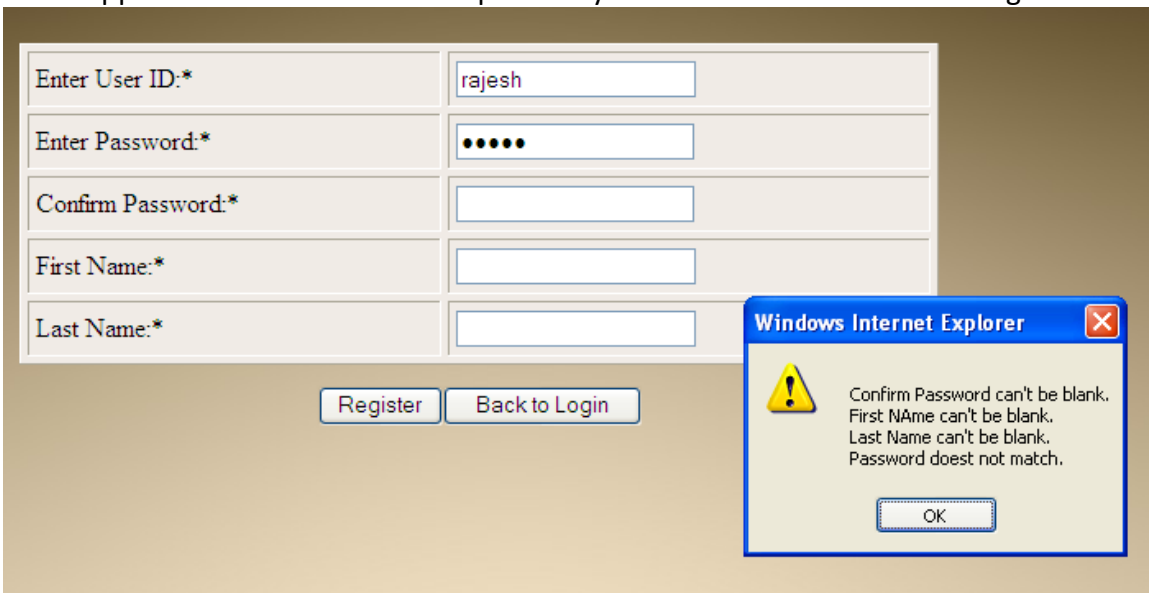
This section will discuss the results of the Test Cases performed. It will elaborate on all the important aspects of the results obtained. It will showcase the results in screenshots taken while running the test cases.

As indicated, Test Cases are formed to test both positive and negative aspects of SIRAT. This means that you pass the correct information and also wrong information and test the tool to check for the output. This is done to check the reliability of the tool.

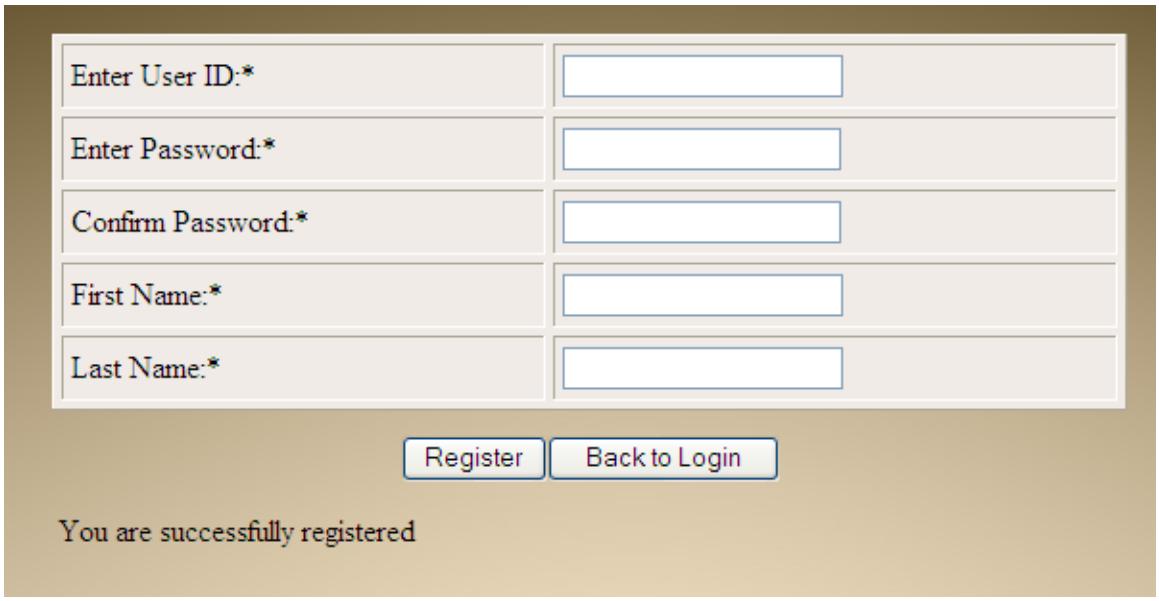
7.5.1 Login and Log out functionality – Testing this functionality, Login/Logout was successful when providing the correct credentials. Furthermore, SIRAT did not allow the user to login when either username or password was not correct, with an alerting message shown below:



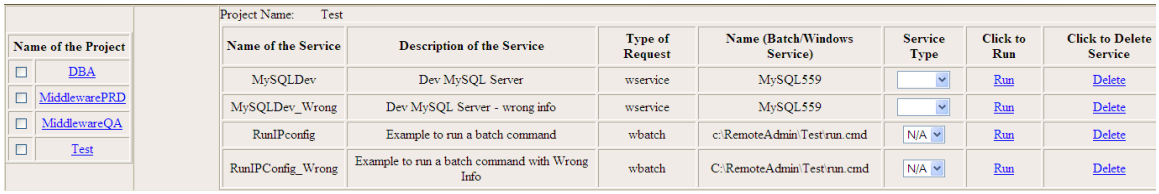
7.5.2 Registration – During registration, no field can be blank. The following shows what happens if a field is left blank. Specifically the user is alerted with a dialog box.



SIRAT provides the user with the following window when registration is successful.



7.5.3 Create Project and Service – The following window shows when a project is successfully created. Once the project is created, a link to the project bearing its name is shown on the left hand side. It also shows when a windows service and a windows batch file with correct and wrong information are created, respectively (the name of the Project is *Test* and the names of the services are *MySQLDev, MySQLDev_Wrong, RunIPConfig* and *RunIPConfig_Wrong*).



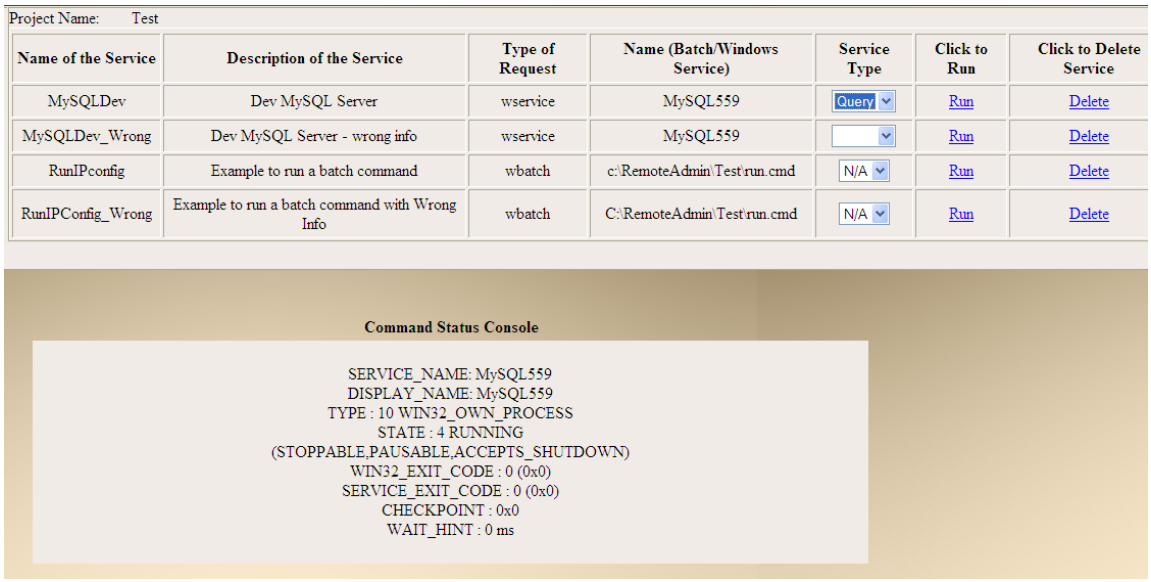
Name of the Project

- DBA
- MiddlewarePRD
- MiddlewareQA
- Test

Project Name: Test

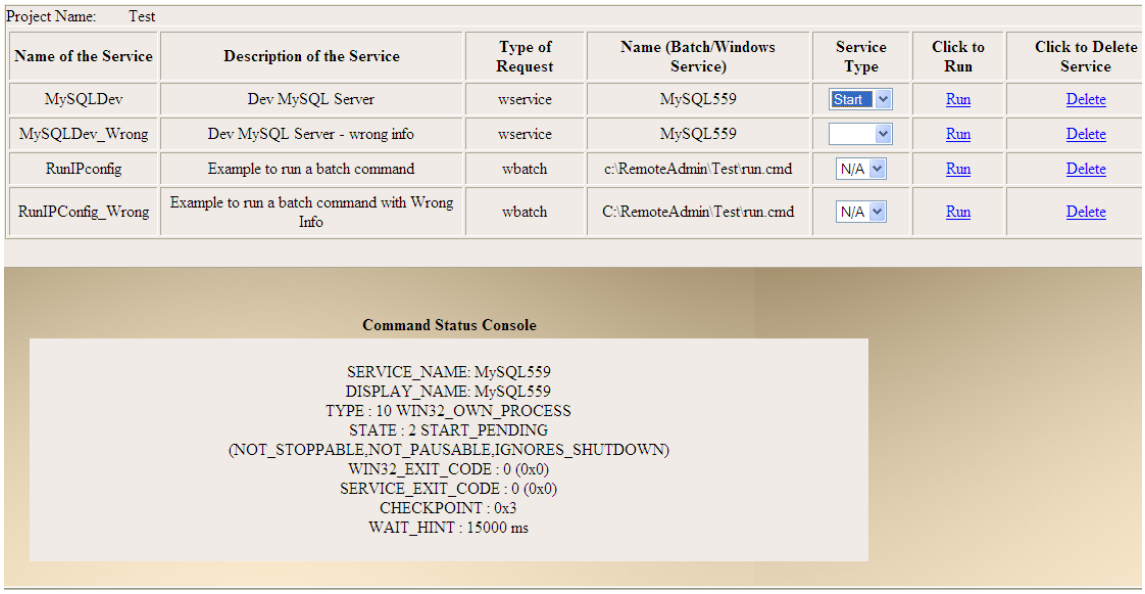
Name of the Service	Description of the Service	Type of Request	Name (Batch/Windows Service)	Service Type	Click to Run	Click to Delete Service
MySQLDev	Dev MySQL Server	wservice	MySQL559		Run	Delete
MySQLDev_Wrong	Dev MySQL Server - wrong info	wservice	MySQL559		Run	Delete
RunIPconfig	Example to run a batch command	wbatch	c:\RemoteAdmin\Test\run.cmd	N/A	Run	Delete
RunIPConfig_Wrong	Example to run a batch command with Wrong Info	wbatch	C:\RemoteAdmin\Test\run.cmd	N/A	Run	Delete

7.5.4 Run a Windows Service with Correct Information (Query/Start/Stop) – Clicked on the Windows Service (MySQLDev) that was created to administer the Dev MySQL database server. Started with querying the service and went on to start and stop the service. All of them worked as expected. In the Command Status Console, check the state to know the status of the windows service.



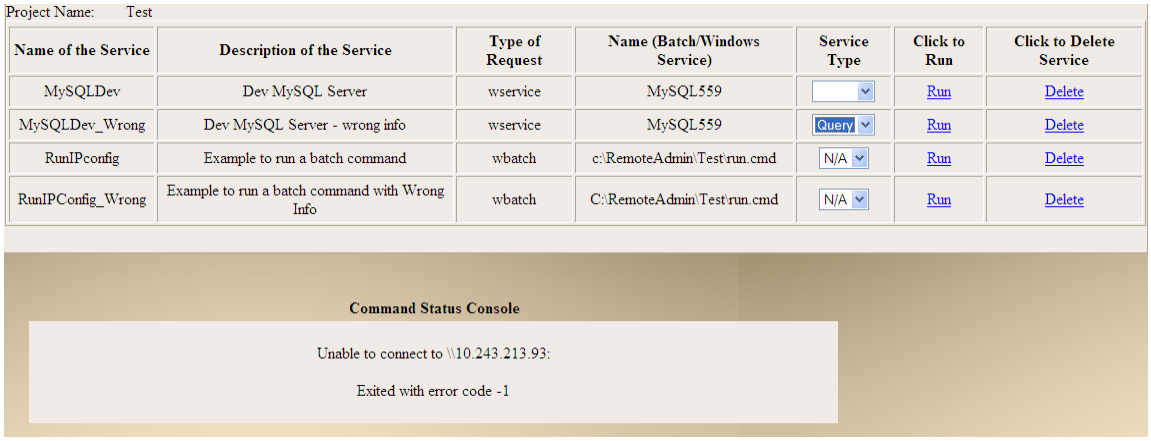
Status 1 - When Querying the Status of the *MySQLDev*, we can see that the State as *RUNNING*.

Status 2 - When executing "stop" for *MySQLDev*, the State turned to *STOP_PENDING*. Query the service again to confirm if the service has stopped or not.



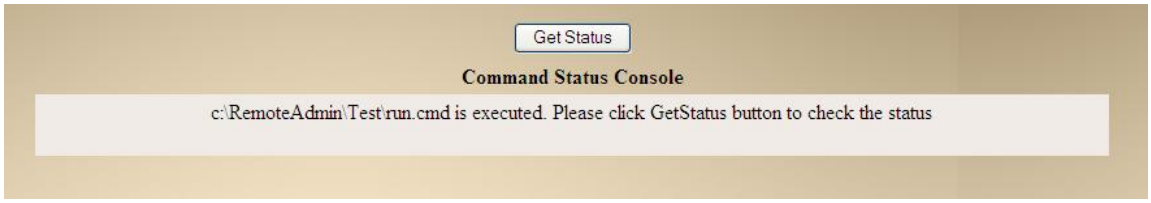
Status 3 - When executing "start" for *MySQLDev*, the State turned to *START_PENDING.* Query the service again to confirm if the service has started or not.

7.5.5 Run a Windows Service with Incorrect Information – Created a windows service (MySQLDev_Wrong) with giving wrong information (incorrect IP Address of the remote machine) and see what could be the output.

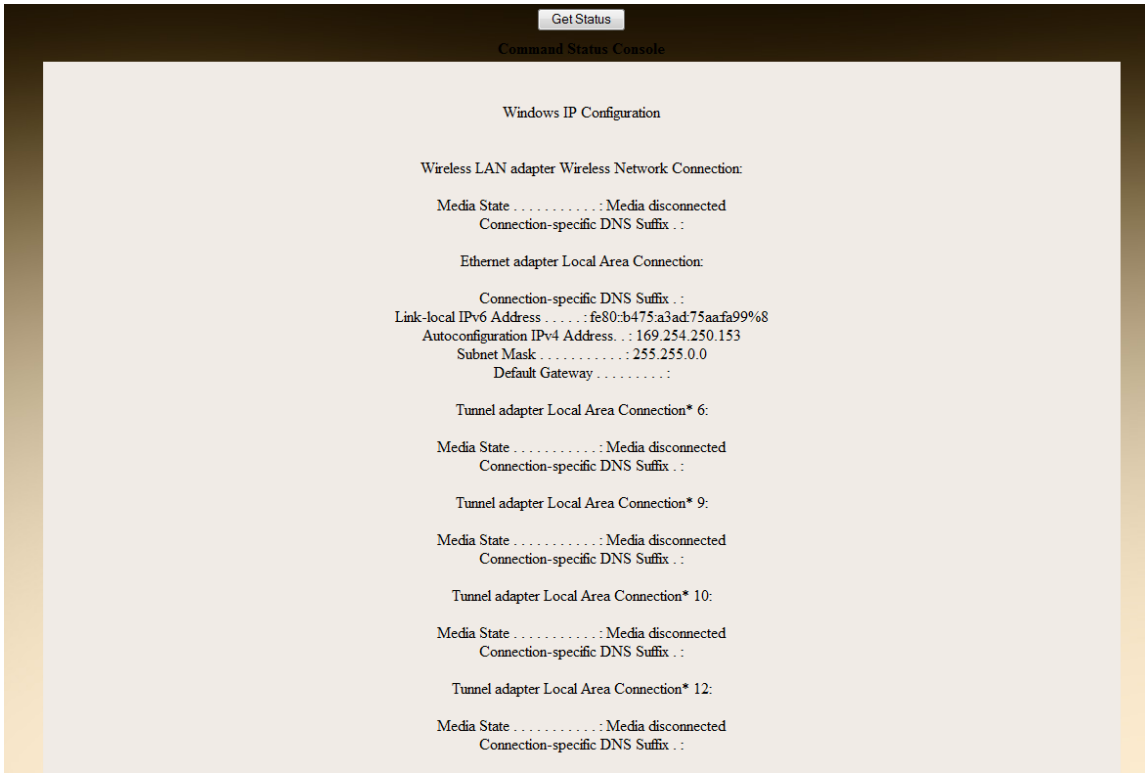


Status 4 – In the Command Status Console it shows that SIRAT was unable to connect to the IP Address supplied.

7.5.6 Run a Windows Batch File with correct Information – To test the functionality of SIRAT with respect to a batch file, RunIPconfig service is created and executed. Note that the batch file needs to reside on the remote machine. On running the batch file, the results show the IP address on the remote box. There are some customizations for running a batch file, which is explained in 5.3.3 section. Please read this section for further clarification on how to create a batch file on the remote machine.

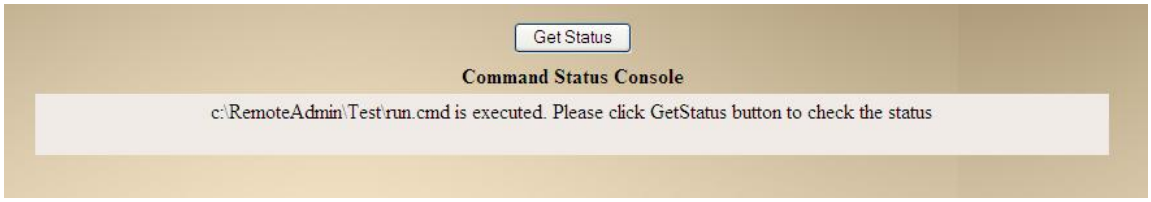

Get Status
Command Status Console
c:\RemoteAdmin\Test\run.cmd is executed. Please click GetStatus button to check the status

The above screen is provided to the user when a batch file is run. To check on the status of the batch file, click on "Get Status" button.


Get Status
Command Status Console
Windows IP Configuration
Wireless LAN adapter Wireless Network Connection:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::b475:a3ad:75aa:fa99%8
Autoconfiguration IPv4 Address. . : 169.254.250.153
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
Tunnel adapter Local Area Connection* 6:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter Local Area Connection* 9:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter Local Area Connection* 10:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter Local Area Connection* 12:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Status 5 – The above screenshot is showing the contents of the log output of RunIPConfig batch file which was executed on the remote machine.

7.5.7 Run a Windows Batch File with Incorrect Information – To show what happens when incorrect information is given, like giving incorrect password or incorrect IP address of the remote box. A service is created called *RunIPconfig_Wrong* with IP Address of the remote machine.


Get Status
Command Status Console
c:\RemoteAdmin\Test\run.cmd is executed. Please click GetStatus button to check the status

The above screen is what you see when you run RunIPconfig_Wrong. In order to see the Status, if the batch file ran successfully or not, you need to click on "***Get Status"*** button.

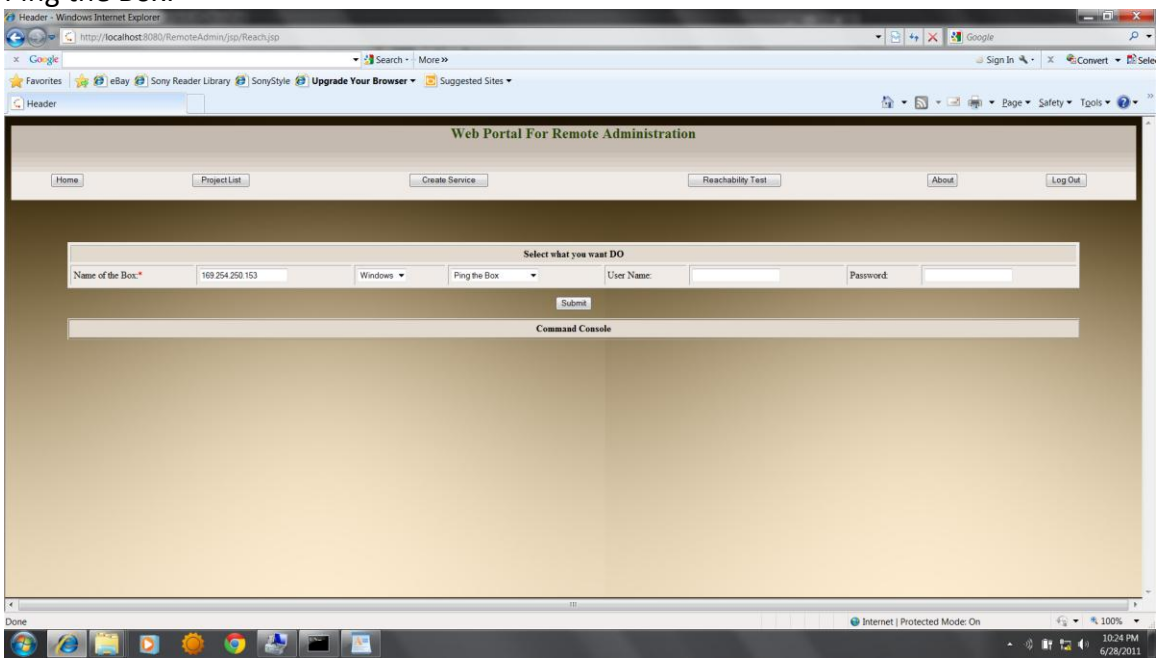
The above message says that the network path is not found. That means SIRAT is unable to reach the remote machine. SIRAT was unable to get to the remote machine as the IP Address supplied is wrong.

7.5.8 Reachability Test

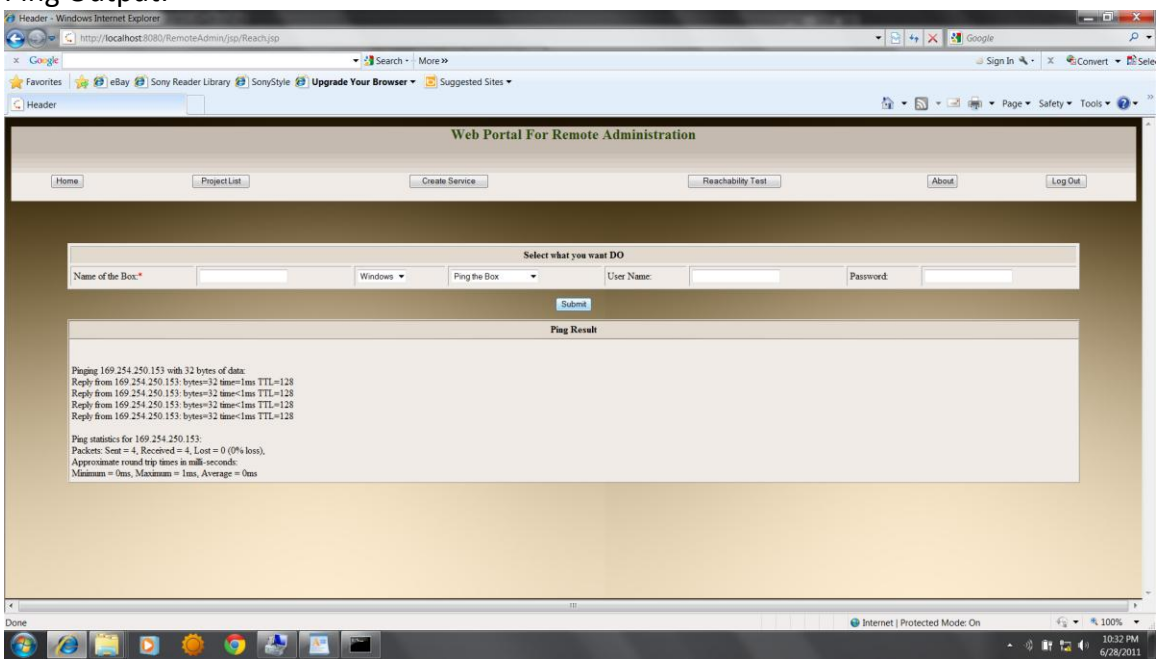
Here we will be testing the following cases with respect to positive and negative input data.

- Ping the Box
- Box Configuration
- Box Memory Usage
- Box Disk Usage
- Box Task Manager

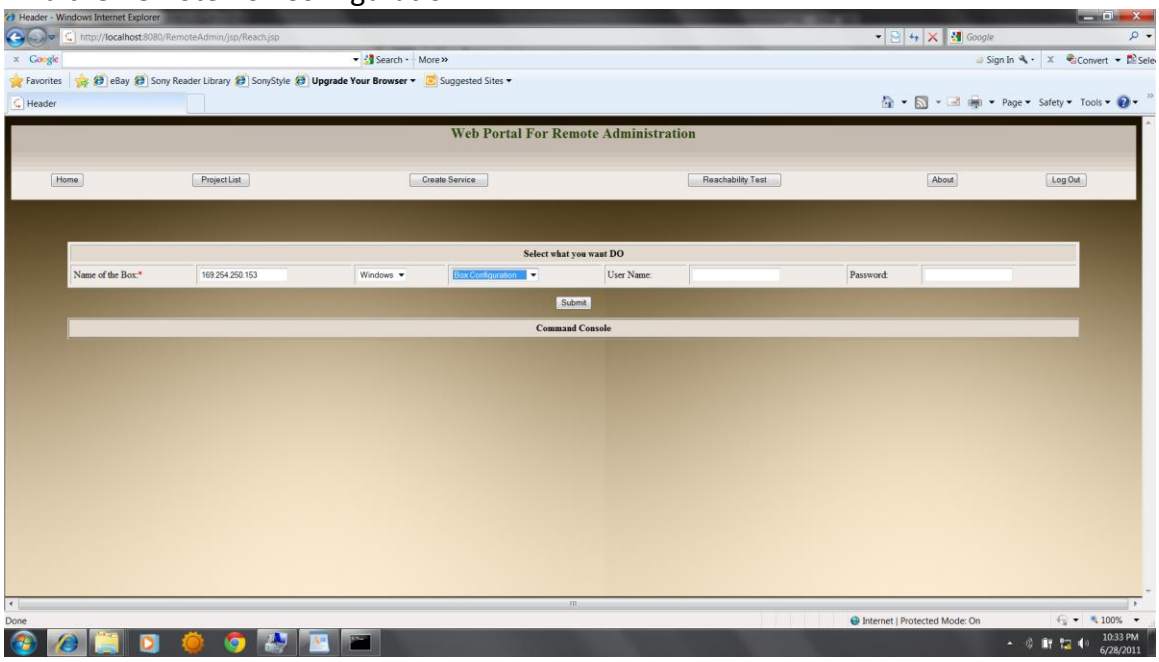
Ping the Box:



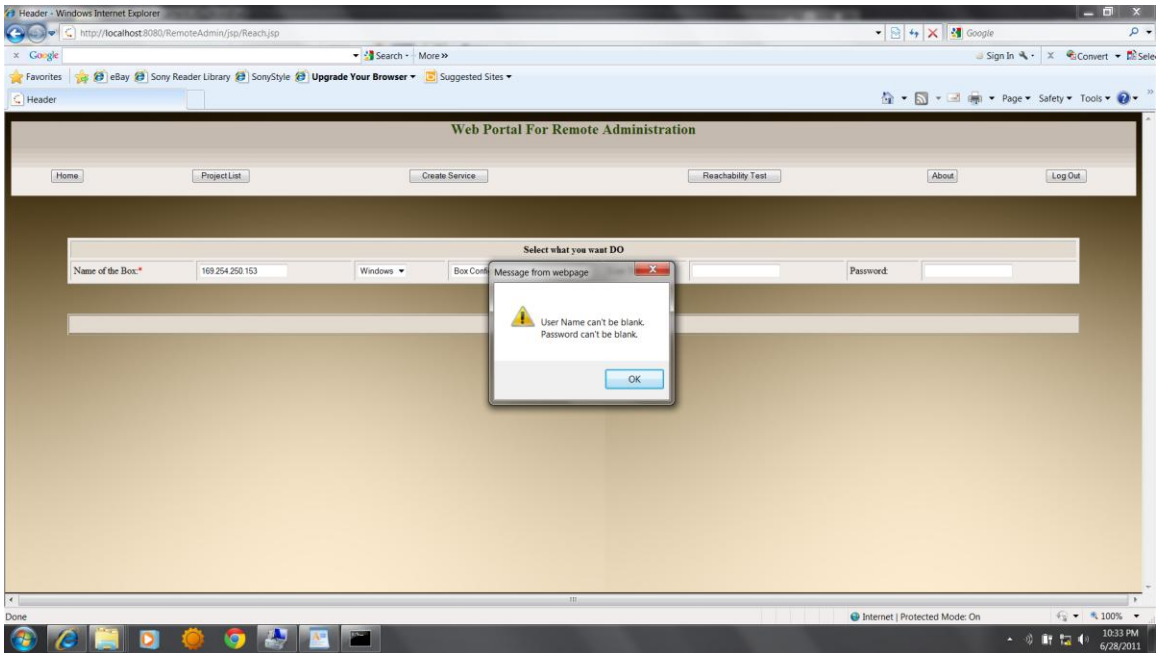
Ping Output:



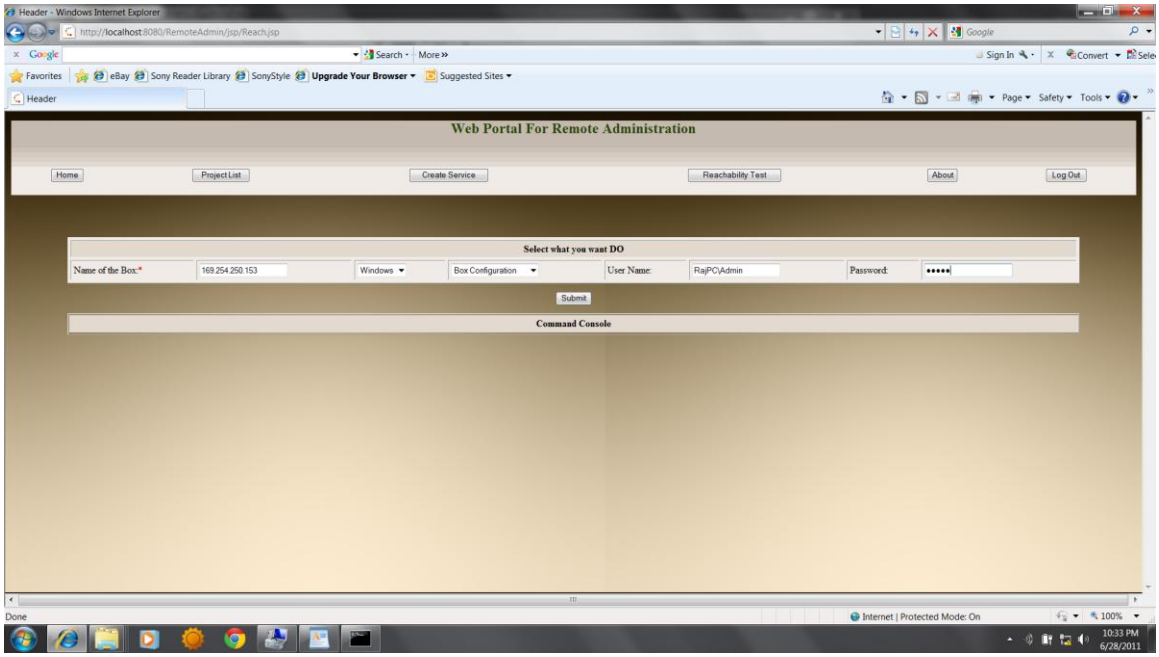
Find the Remote Box Configuration:



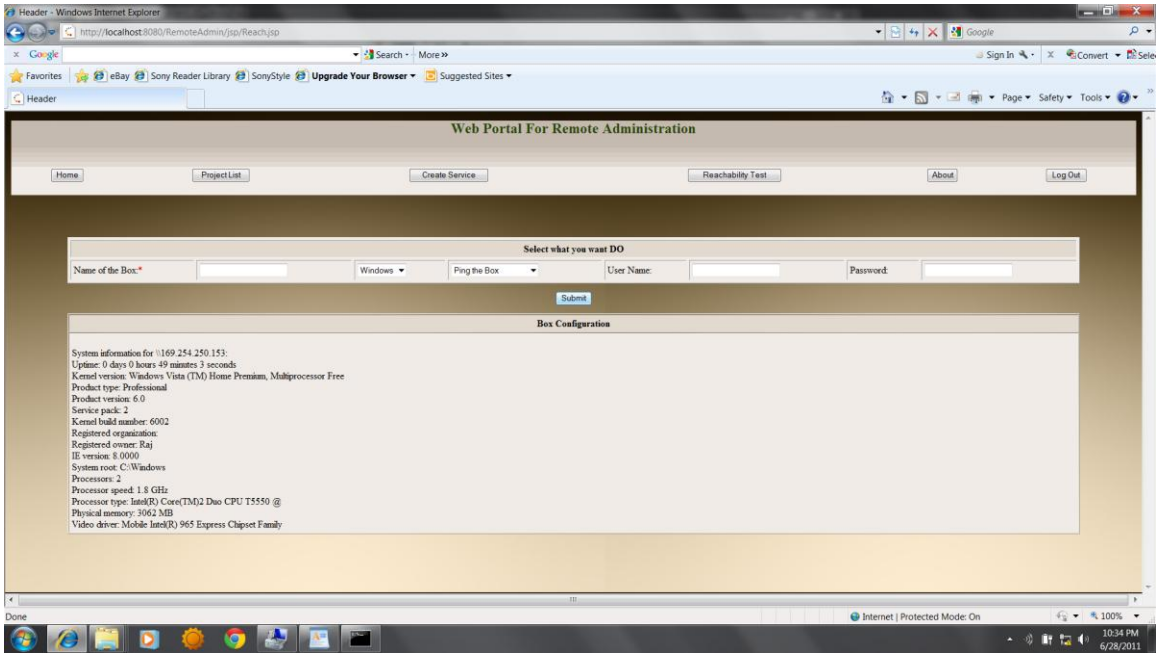
Note that to run this function, the userid/password of the remote box is needed. Otherwise a dialogue box will open and it will ask for the needed credentials.



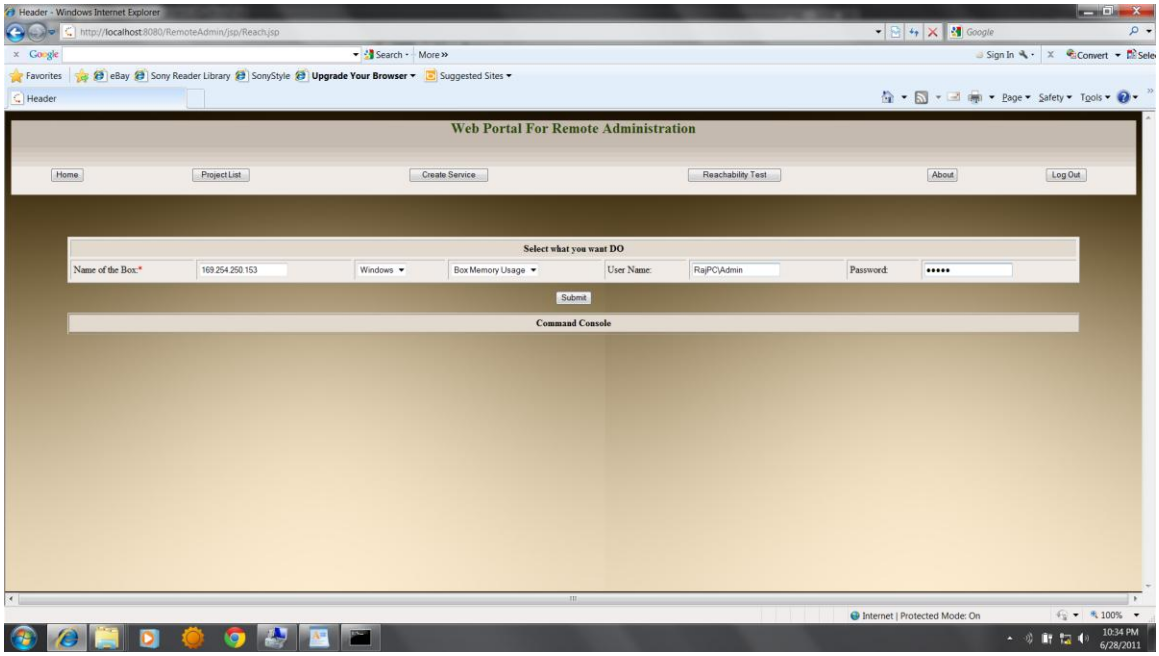
Find the Remote Machine Configuration:



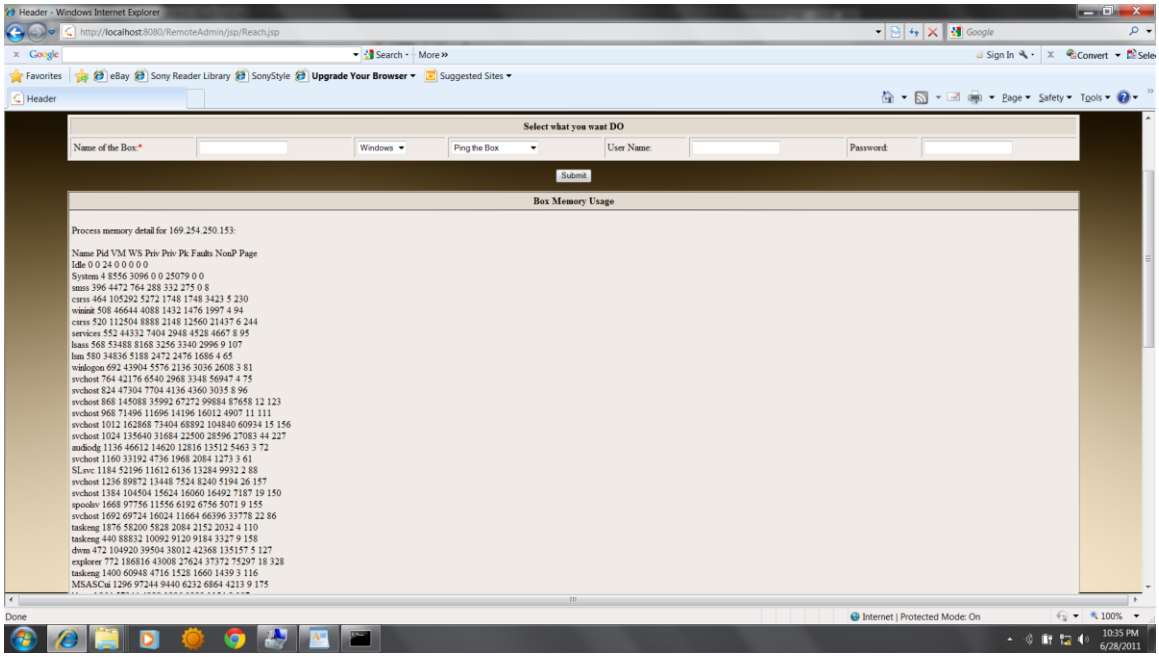
Output of the Remote Machine Configuration:



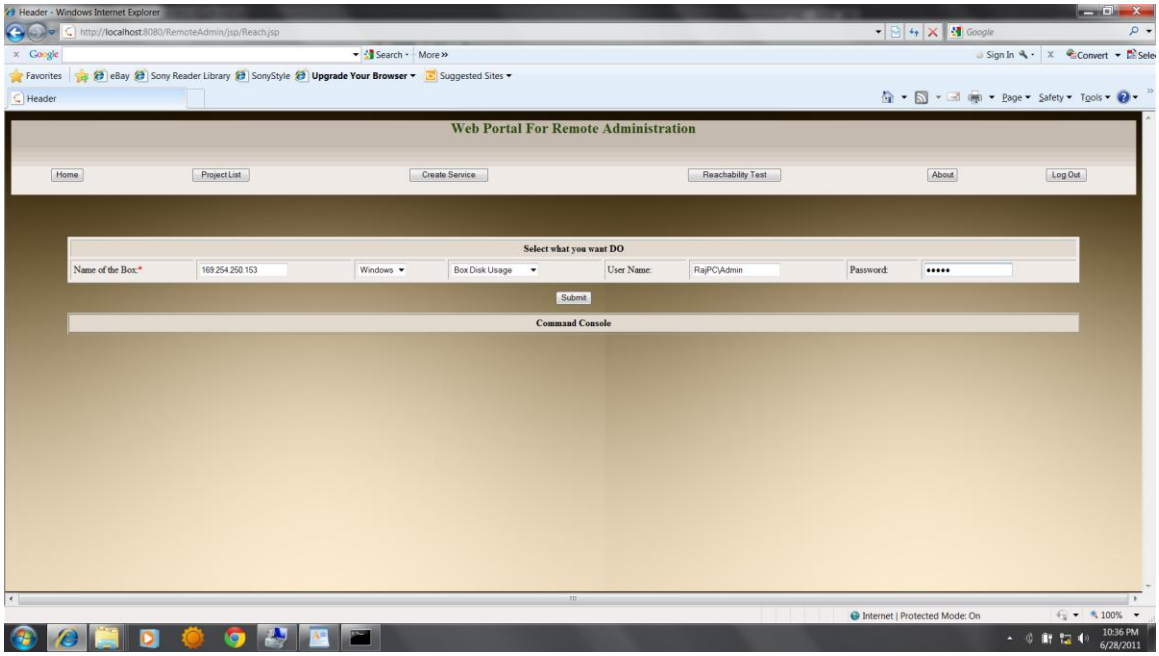
Find the Memory Usage of the Remote Machine:



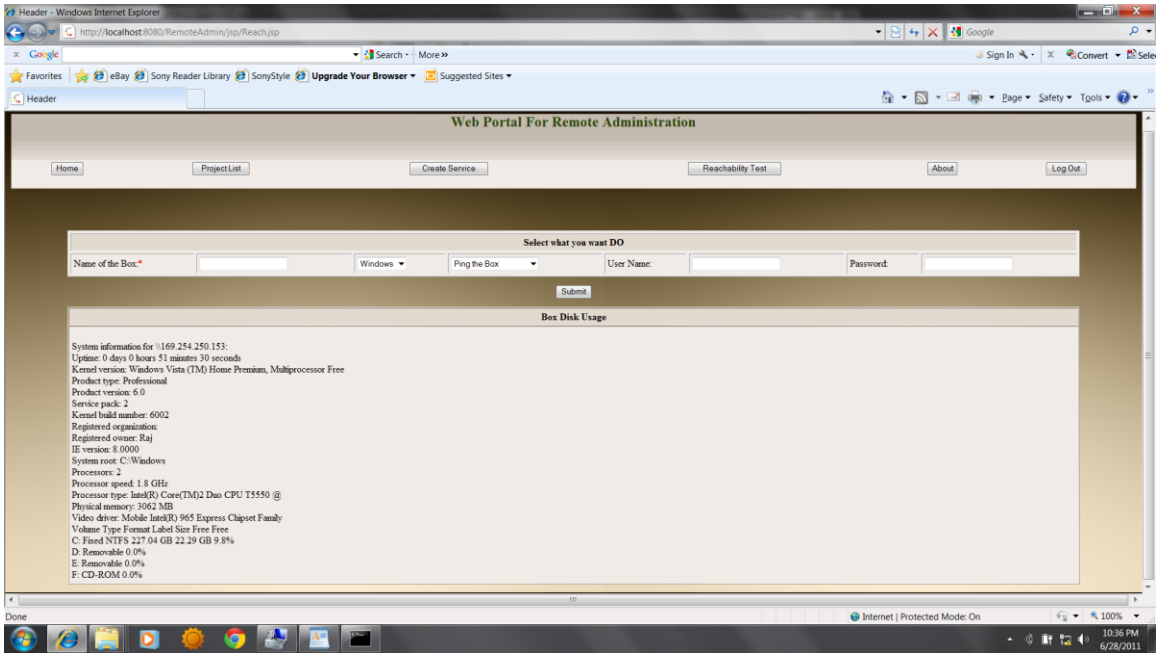
Output of the Memory Usage of the Remote Machine:



Find the Disk usage of the Remote machine:



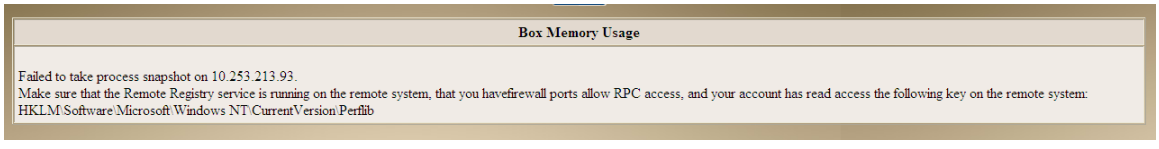
Output of the Disk usage of the Remote machine:



Ping Output, Given Wrong Destination:

Ping Result
Pinging 10.253.213.93 with 32 bytes of data:
Request timed out.
Reply from 192.168.36.252: Destination host unreachable.
Reply from 192.168.36.252: Destination host unreachable.
Reply from 192.168.36.252: Destination host unreachable.
Ping statistics for 10.253.213.93:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Output of Remote Machine Memory, When Given Wrong Destination IP Address:

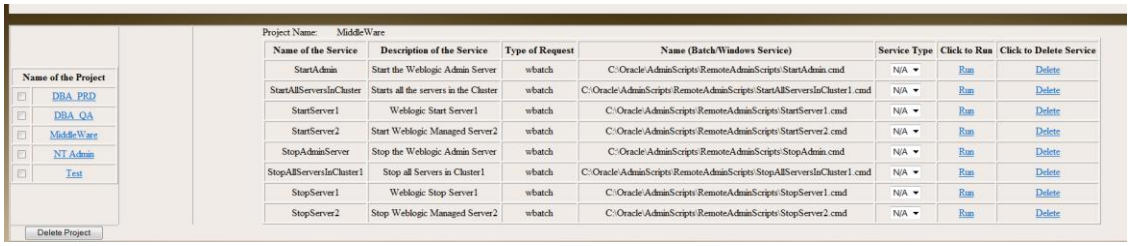

Box Memory Usage
Failed to take process snapshot on 10.253.213.93.
Make sure that the Remote Registry service is running on the remote system, that you havefirewall ports allow RPC access, and your account has read access the following key on the remote system:
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Perflib

7.6 Example: SIRAT Integration with Oracle Weblogic Application Server

The main purpose of building a tool such as SIRAT is to assist administrators in their daily work. In real world IT, administrators have to work on different technologies and they are required to do repetitive tasks on different remote machines. As indicated, this tool is prepared so that it can integrate with these technologies to perform repetitive tasks in an efficient and timely manner.

As an example, a project is created in SIRAT where all the services of Weblogic to be administered are provided. The screen shots below show some of the output when the services are run from SIRAT. The output shows successful start and stop of a domain. The output also shows successful gathering of Weblogic metrics to check on the status of the domain.

StartAdmin and StopAdminServer services in Middleware project are used to start and stop Weblogic Admin Server. This is a simple showcase scenario on how to integrate SIRAT with Weblogic Administration.



Name of the Project

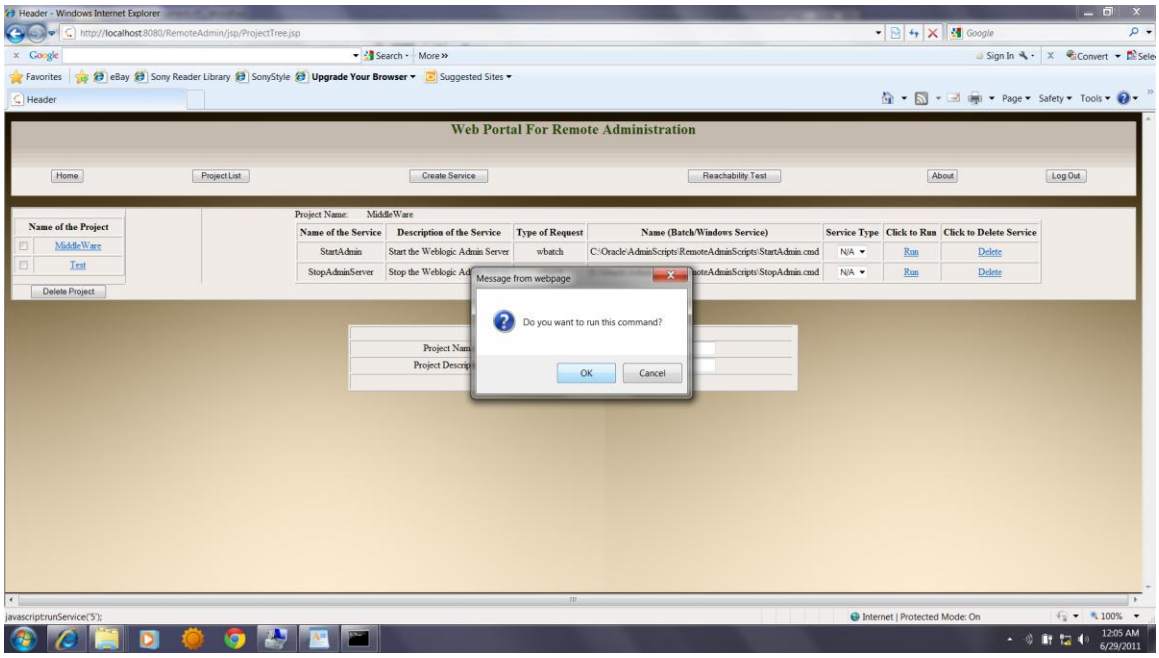
- DBA_PRD
- DBA_QA
- MiddleWare
- NT Admin
- Test

Delete Project

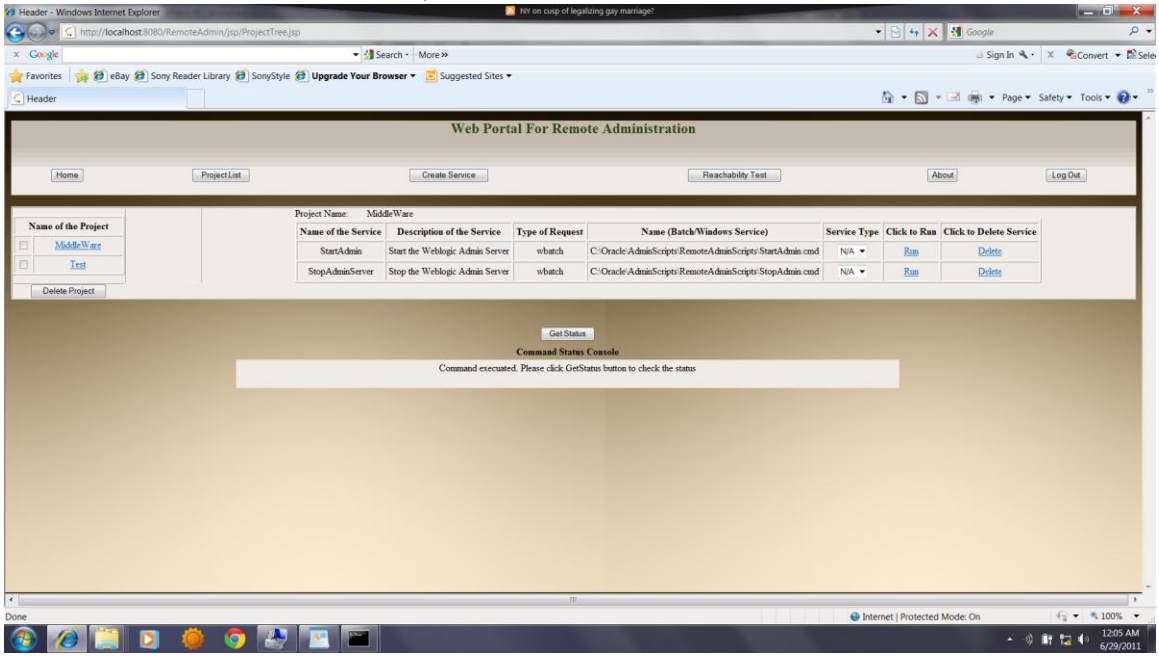
Project Name: MiddleWare

Name of the Service	Description of the Service	Type of Request	Name (Batch/Windows Service)	Service Type	Click to Run	Click to Delete Service
StartAdmin	Start the Weblogic Admin Server	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StartAdmin.cmd	N/A	Run	Delete
StartAllServersInCluster	Starts all the servers in the Cluster	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StartAllServersInCluster1.cmd	N/A	Run	Delete
StartServer1	Weblogic Start Server1	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StartServer1.cmd	N/A	Run	Delete
StartServer2	Start Weblogic Managed Server2	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StartServer2.cmd	N/A	Run	Delete
StopAdminServer	Stop the Weblogic Admin Server	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StopAdmin.cmd	N/A	Run	Delete
StopAllServersInCluster1	Stop all Servers in Cluster1	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StopAllServersInCluster1.cmd	N/A	Run	Delete
StopServer1	Weblogic Stop Server1	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StopServer1.cmd	N/A	Run	Delete
StopServer2	Stop Weblogic Managed Server2	wbatch	C:\Oracle\AdminScripts\RemoteAdminScripts\StopServer2.cmd	N/A	Run	Delete

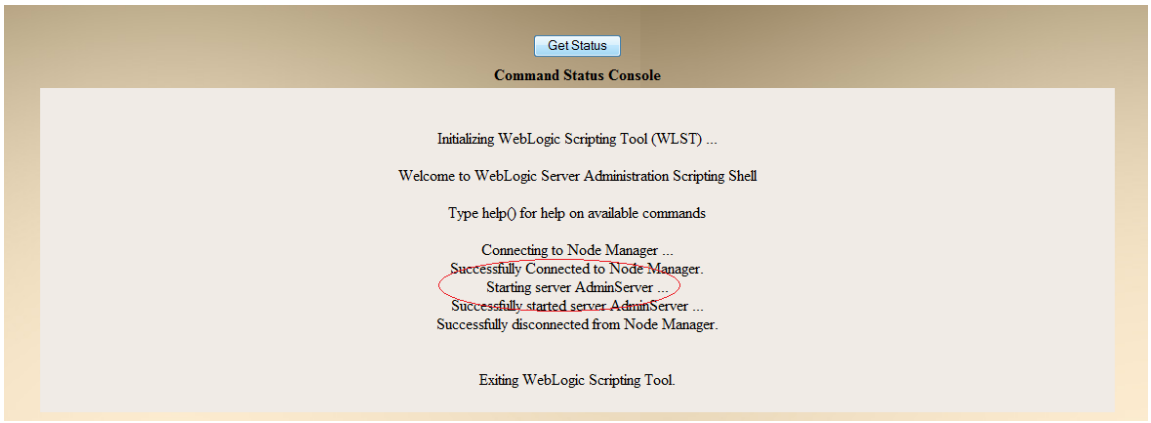
In the screenshot above, when we click on “Run” to start the service *StartAdmin*, SIRAT will ask you for confirmation, as shown in below screenshot. Once confirmed, SIRAT will execute the remote batch file which starts the Weblogic Admin server on the remote machine.



To check the status of execution, click on “Get Status” button.

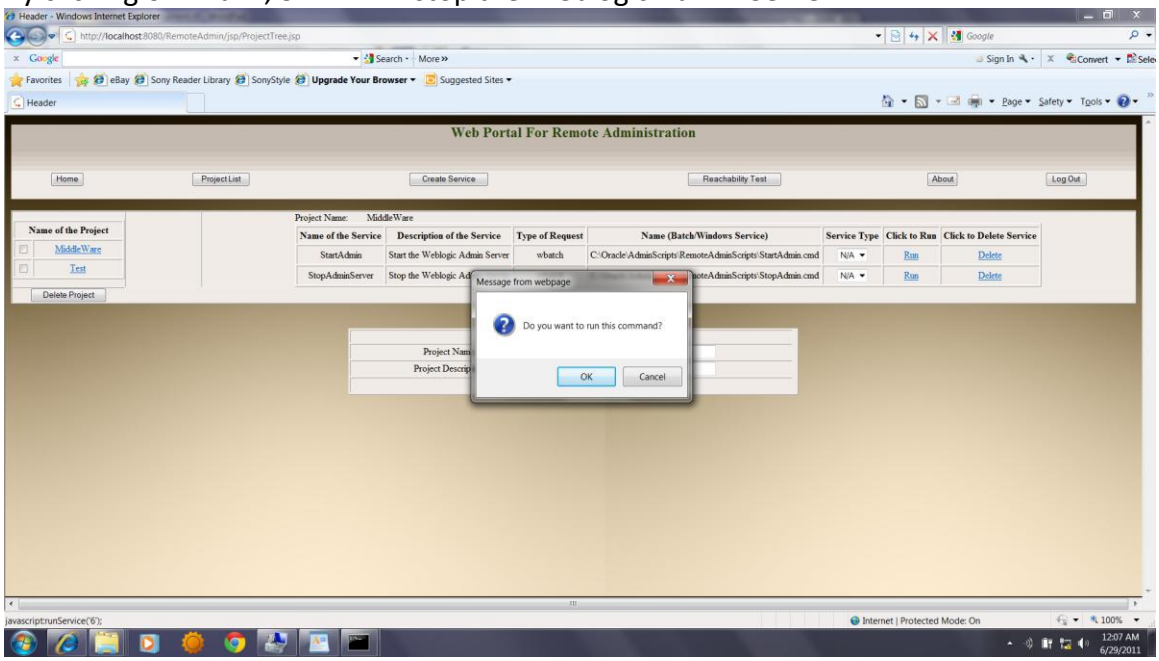


SIRAT will read the log output of the batch file executed on the remote machine and states that it has started the Weblogic Admin Server successfully.

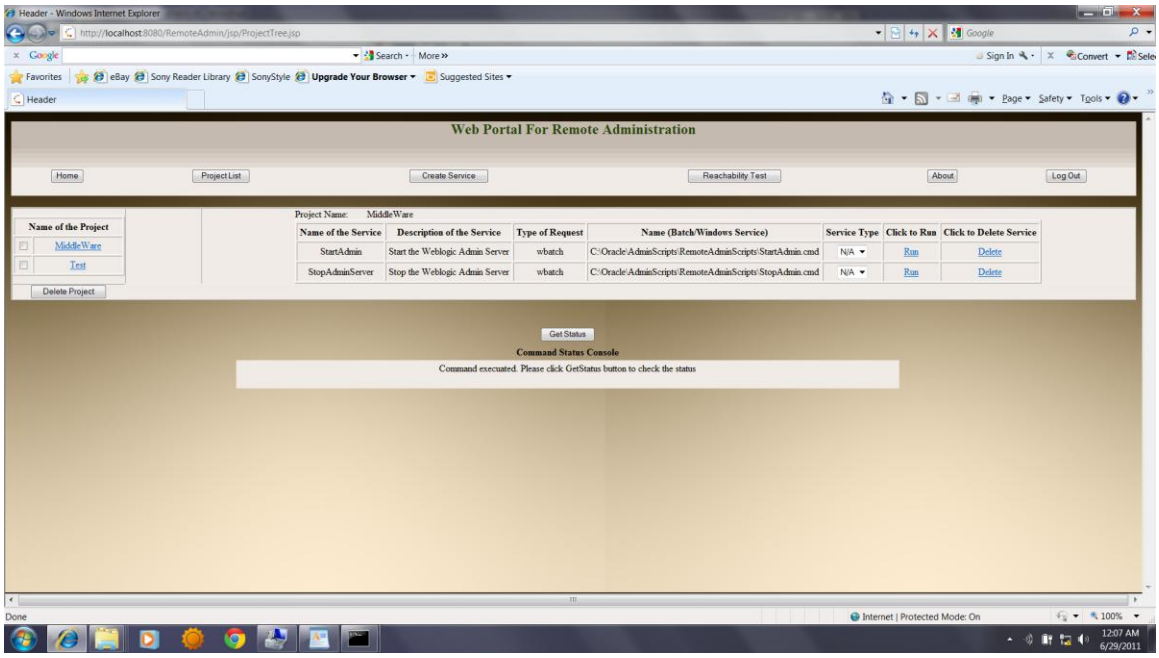

Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to Node Manager ...
Successfully Connected to Node Manager.
Starting server AdminServer ...
Successfully started server AdminServer ...
Successfully disconnected from Node Manager.
Exiting WebLogic Scripting Tool.

Similarly the next three screens shots show how to stop the Weblogic Admin Server followed by the status of the execution that can be checked.

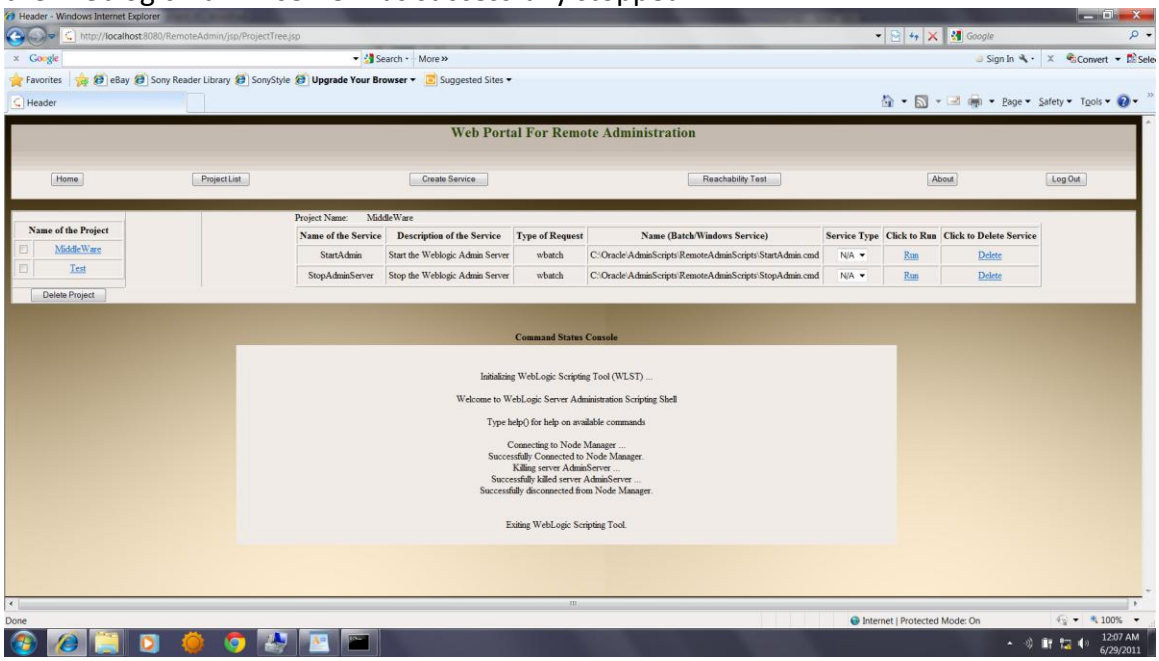
By clicking on “Run”, SIRAT will stop the Weblogic Admin Server.



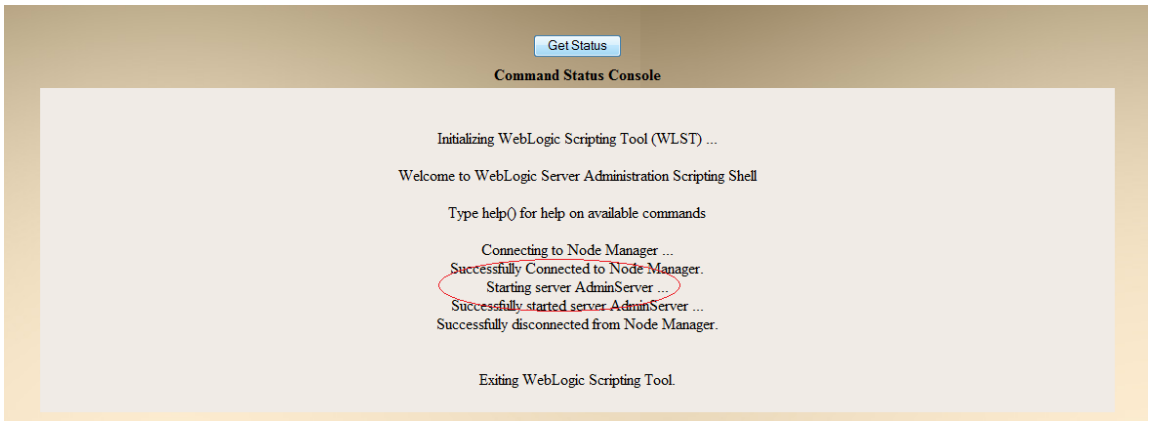
The screenshot below indicates that the command is executed and will ask you to click on “Get Status” button to get the status on the command executed.

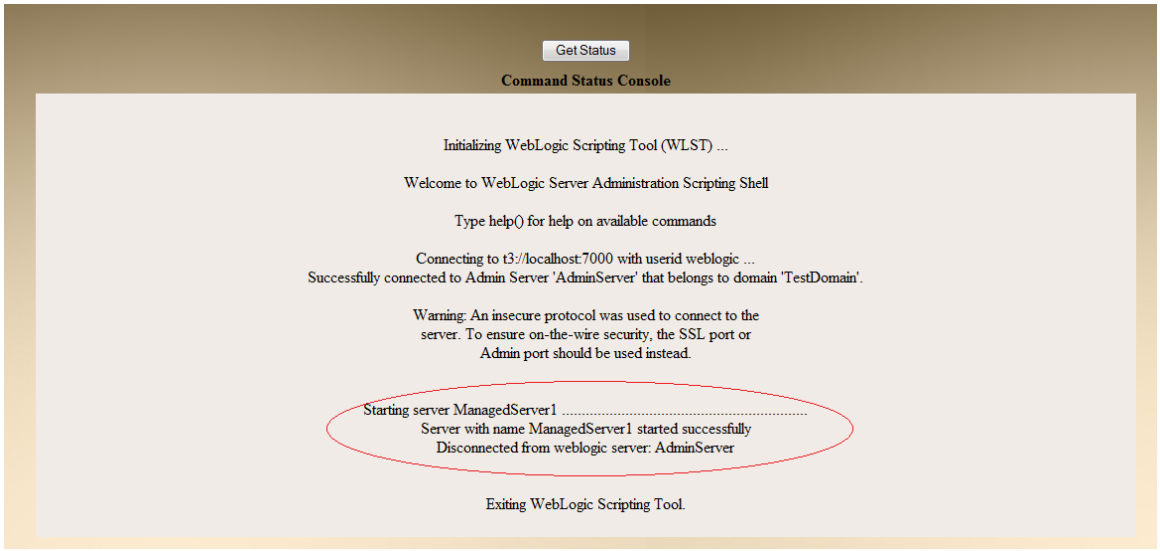


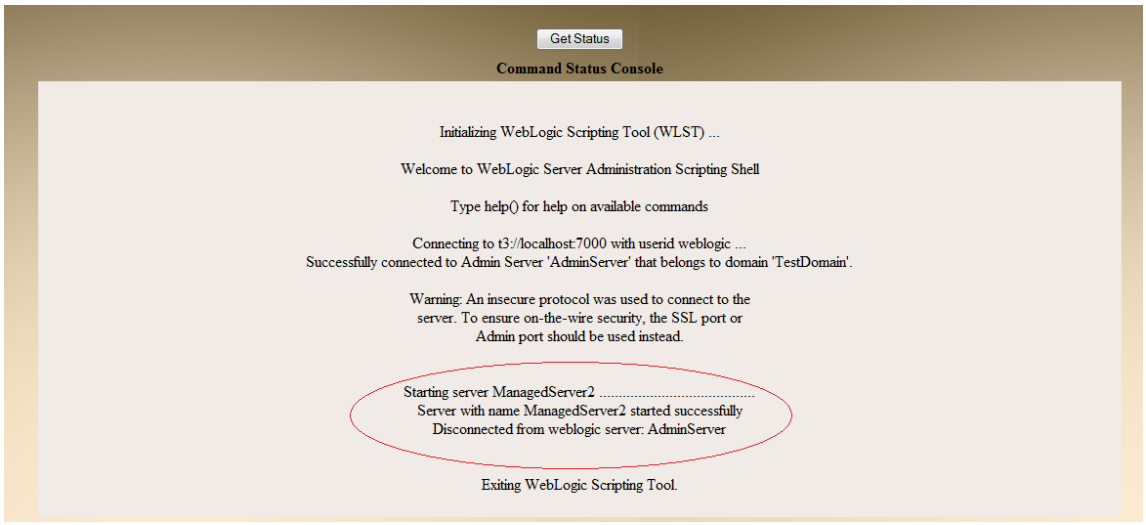
The screenshot below shows the output of the batch execution. The screen shows that the Weblogic Admin server has successfully stopped.

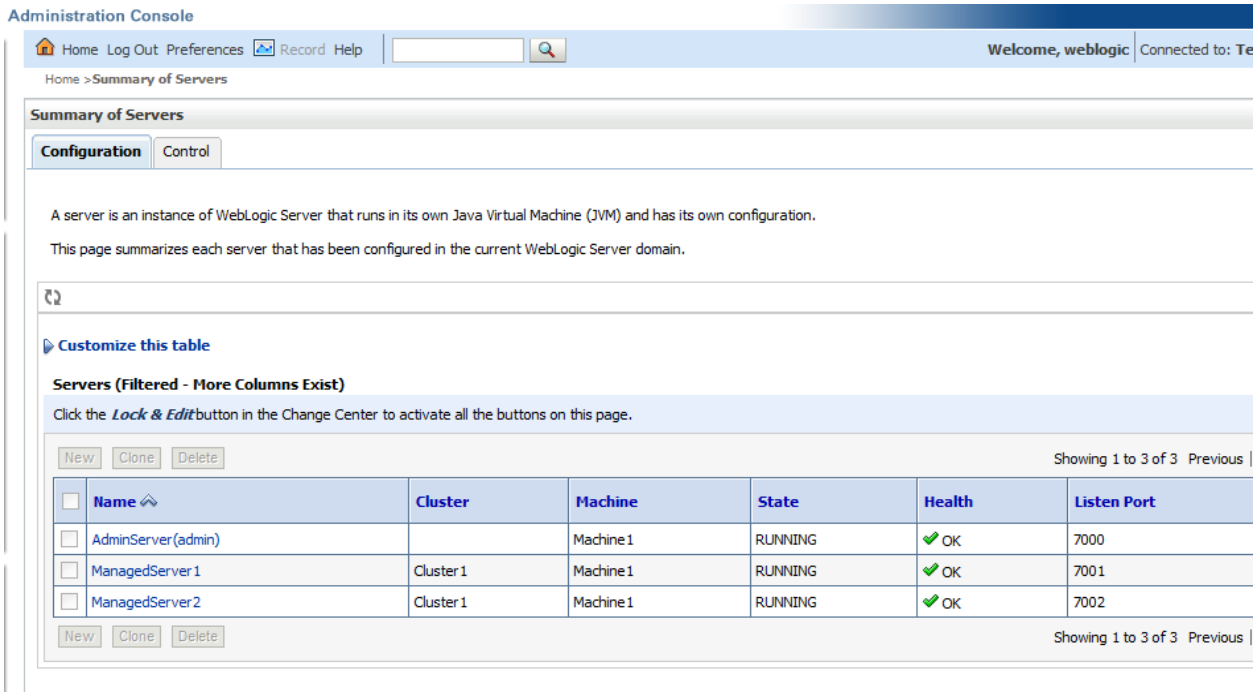


The next eight screen shots shown below are the results of starting the entire Weblogic Domain using SIRAT.

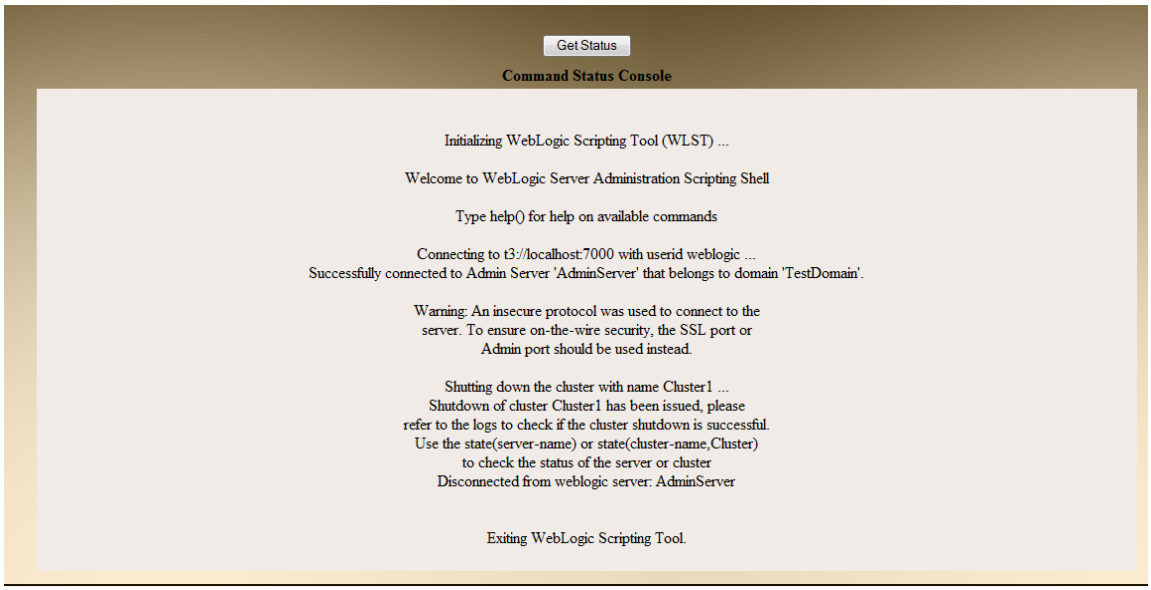

Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to Node Manager ...
Successfully Connected to Node Manager.
Starting server AdminServer ...
Successfully started server AdminServer ...
Successfully disconnected from Node Manager.
Exiting WebLogic Scripting Tool.

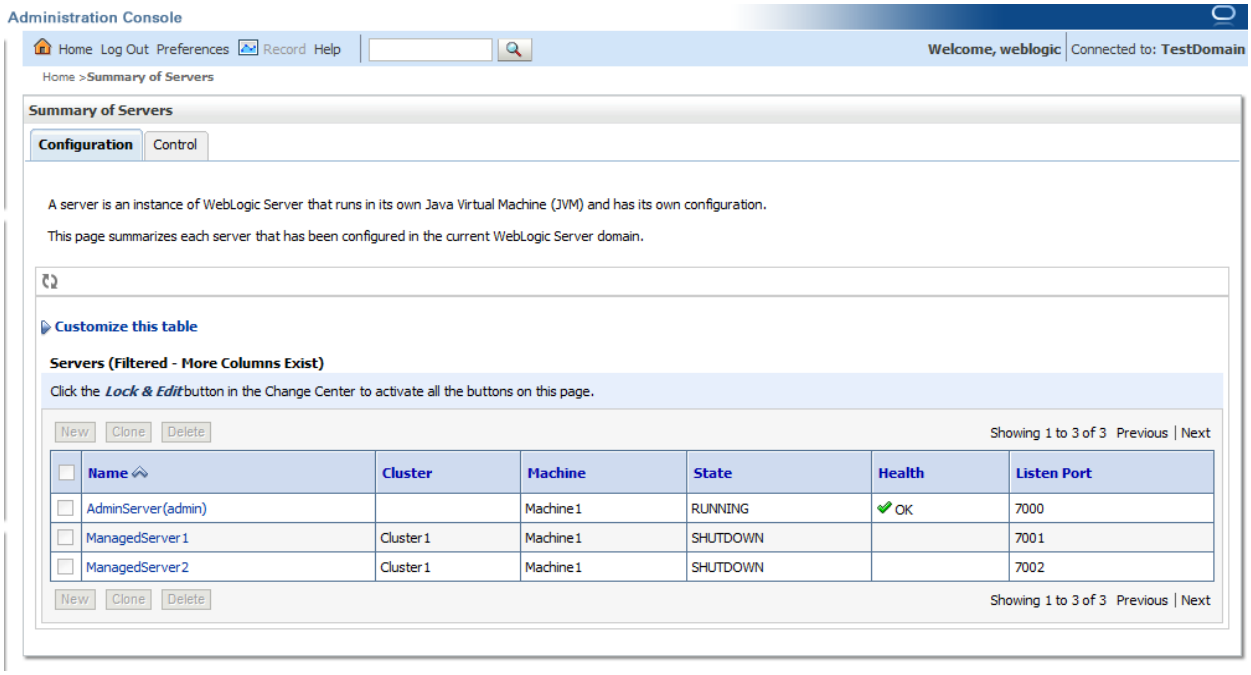

Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to t3://localhost:7000 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain 'TestDomain'.
Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.
Starting server ManagedServer1 .....
Server with name ManagedServer1 started successfully
Disconnected from weblogic server: AdminServer
Exiting WebLogic Scripting Tool.


Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to t3://localhost:7000 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain 'TestDomain'.
Warning: An insecure protocol was used to connect to the server. To ensure on-the-wire security, the SSL port or Admin port should be used instead.
Starting server ManagedServer2 .....
Server with name ManagedServer2 started successfully
Disconnected from weblogic server: AdminServer
Exiting WebLogic Scripting Tool.

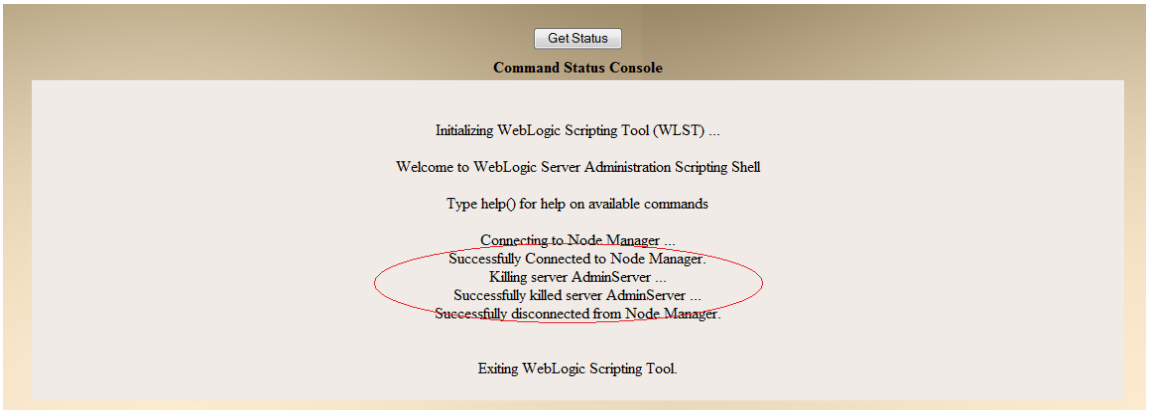

Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: Te
Home >Summary of Servers
Summary of Servers
Configuration Control
A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.
This page summarizes each server that has been configured in the current WebLogic Server domain.
Customize this table
Servers (Filtered - More Columns Exist)
Click the Lock & Edit button in the Change Center to activate all the buttons on this page.
New Clone Delete
Showing 1 to 3 of 3 Previous
Name Cluster Machine State Health Listen Port
AdminServer(admin) Machine1 RUNNING OK 7000
ManagedServer1 Cluster1 Machine1 RUNNING OK 7001
ManagedServer2 Cluster1 Machine1 RUNNING OK 7002
New Clone Delete
Showing 1 to 3 of 3 Previous

The above screen is the Weblogic Administration console where it shows that the Admin Server and the two managed servers have started successfully by using SIRAT.

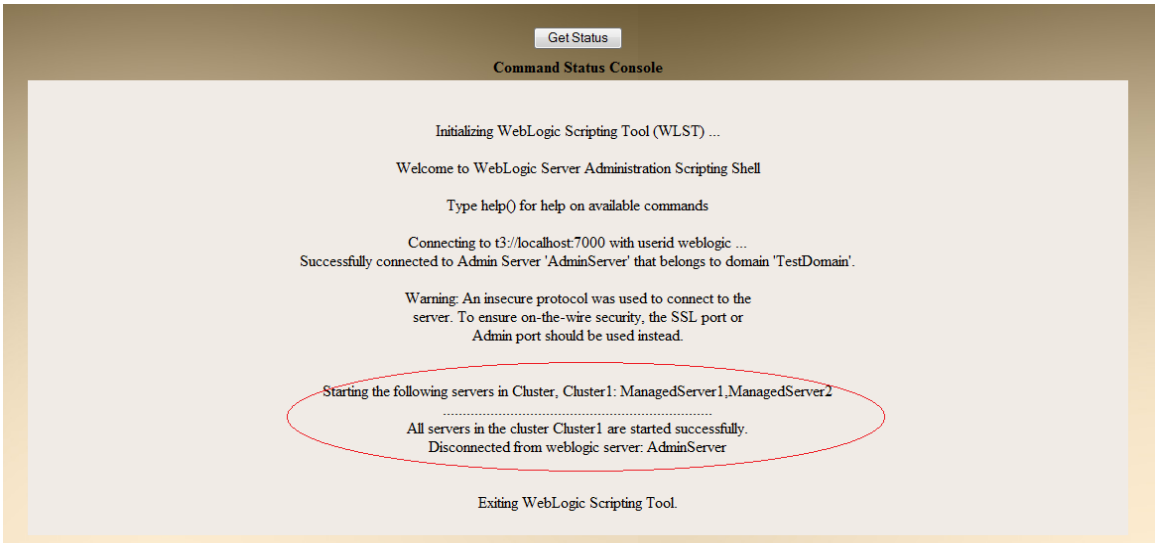

Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to t3://localhost:7000 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain 'TestDomain'.
Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.
Shutting down the cluster with name Cluster1 ...
Shutdown of cluster Cluster1 has been issued, please
refer to the logs to check if the cluster shutdown is successful.
Use the state(server-name) or state(cluster-name,Cluster)
to check the status of the server or cluster
Disconnected from weblogic server: AdminServer
Exiting WebLogic Scripting Tool.


Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: TestDomain
Home >Summary of Servers
Summary of Servers
Configuration Control
A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.
This page summarizes each server that has been configured in the current WebLogic Server domain.
Customize this table
Servers (Filtered - More Columns Exist)
Click the Lock & Edit button in the Change Center to activate all the buttons on this page.
New Clone Delete
Showing 1 to 3 of 3 Previous | Next
Name Cluster Machine State Health Listen Port
AdminServer(admin) Machine1 RUNNING OK 7000
ManagedServer1 Cluster1 Machine1 SHUTDOWN 7001
ManagedServer2 Cluster1 Machine1 SHUTDOWN 7002
New Clone Delete
Showing 1 to 3 of 3 Previous | Next

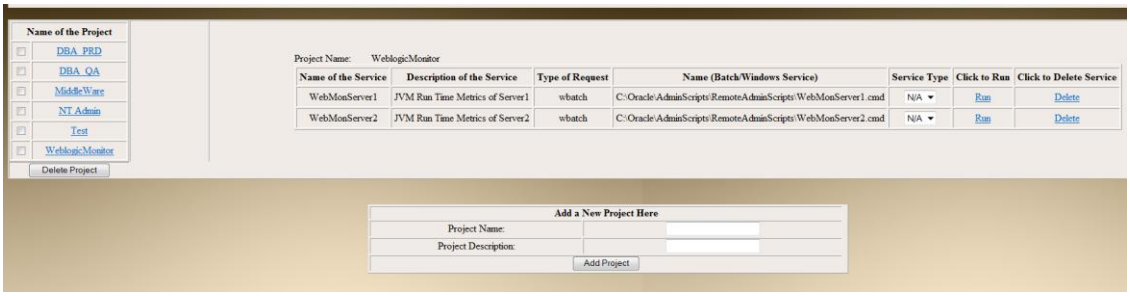
The above screen is the Weblogic Administration console where it shows that the two managed servers have been stopped by executing StopAllServersInCluster1 batch file on the remote machine using SIRAT. This has stopped the two weblogic managed servers.


Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to Node Manager ...
Successfully Connected to Node Manager.
Killing server AdminServer ...
Successfully killed server AdminServer ...
Successfully disconnected from Node Manager.
Exiting WebLogic Scripting Tool.

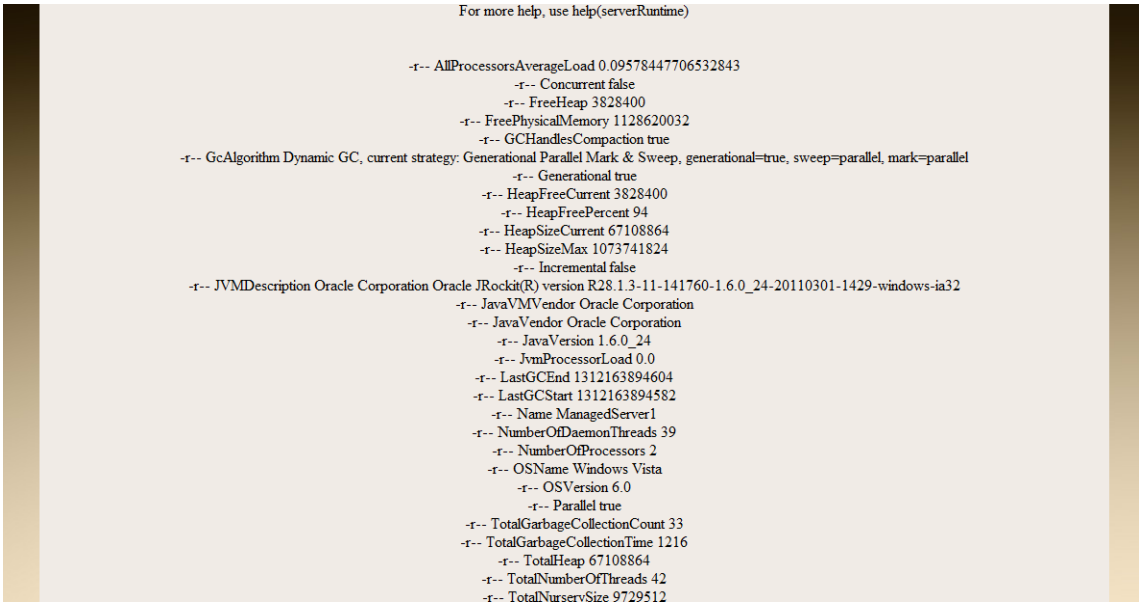
The screenshot shown below is the result of running StopAllServersInCluster1 service. It stops all the servers in cluster1 (i.e. Server1 and Server2).


Get Status
Command Status Console
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to t3://localhost:7000 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain 'TestDomain'.
Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.
Starting the following servers in Cluster, Cluster1: ManagedServer1,ManagedServer2
All servers in the cluster Cluster1 are started successfully.
Disconnected from weblogic server: AdminServer
Exiting WebLogic Scripting Tool.

WeblogicMonitor project was created to keep all the weblogic monitoring scripts. Here two services have been created (the screenshot shown below) to monitor JVM Runtime metrics on server1 and server2.



The result shown below is what you see when the service *WebMonServer1* was executed from SIRAT. It shows all the JVM Runtime parameters of Server1.



7.7 Performance Testing

Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or a device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. Qualitative and quantitative attributes such as reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing.

Performance testing can verify that a system meets the specifications claimed by its manufacturer or vendor. The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability.

SIRAT was run for several hours under observation; the most important parameters such as memory and CPU of the machine were monitored. There was no degrading effect on the machine and it used minimal resources for functioning. The main components of SIRAT that need review from a performance angle are the utilization of JBOSS and MySQL server. To check on the status of the resource usage of these processes, memory and CPU usage of JBOSS and MySQL server are recorded periodically.

The following shows some configuration information for a machine on which SIRAT is installed:

Hardware: Laptop
MS Windows XP Professional
Intel Core 2 Duo 2.00 GHz
2 GB Ram

The following figure shows the resources used during those hours:

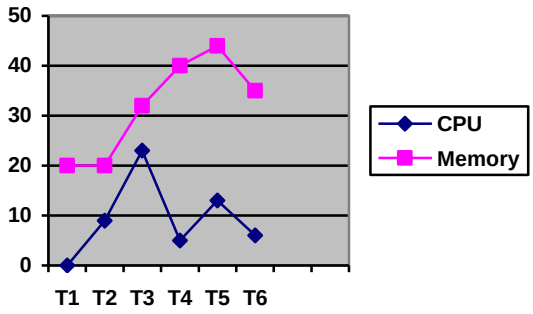


Figure 8 - JBOSS Application Server resource utilization chart.

On X-axis T1, T2, …, are specific intervals. Memory usage is in bytes. To show the actual memory usage, 384000 bytes should be added to the Y-axis values. It is done this way to fit the CPU and memory utilization in one chart and make the chart easier to read. The CPU percentage is the value of java.exe process at that particular event. If average is taken then the average CPU value is very negligible.

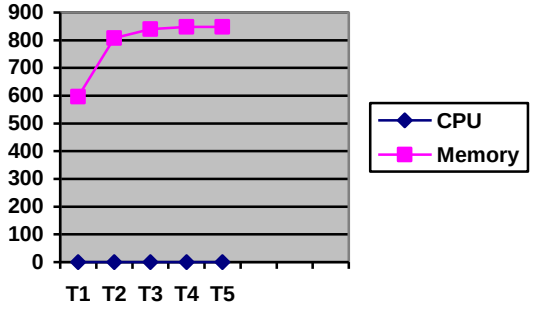


Figure 9 - MySQL Server resource utilization chart.

In Figure 9, T1, …, are specific time intervals. Memory is in bytes. For a similar reason, to find the memory usage, 12000 bytes need to be added to the memory values. The

CPU percentage is the value of *mysql.exe* process at that particular event. If average CPU is considered than again it is negligible.

8 Conclusions

8.1 Summary

The overall effect of SIRAT is to reduce cost and increase productivity. One important benefit of SIRAT is to make daily real world IT administrative activities easier. It has been shown that SIRAT is capable of executing activities on remote boxes from a single machine. In real world IT, there are usually a large number of administrative activities such as Middleware Administration, System Administration, Network Administration, etc. For each of these responsibilities there are normally some daily repetitive activities. Therefore, SIRAT can be of great practical use. Once the setup of the commands is done, anyone with little technical experience ought to be able to use the tool, thus enabling people with limited technical skills to use SIRAT efficiently.

The tool can also be used for monitoring metrics of technologies such as Weblogic Application Server. This is a great advantage showing that SIRAT can be integrated with various technologies. Because of flexibility of SIRAT, it is expected that the tool can be expanded to control the entire production, QA and Development environments from a single interface. SIRAT can be installed as a standalone tool or can be hosted so that all the departments can use it from a single installation.

Numerous tests have been performed on SIRAT to test the reliability of SIRAT - SIRAT is a very reliable tool. Finally, the technologies used to build SIRAT are licensed freeware, so there is no concern about being charged for license fees.

8.2 Problems Encountered and Solved

While working on building SIRAT, a number of issues and challenges were encountered. By the help of advisors and with some research these issues have been resolved. Some of the initial challenges are highlighted below:

- How to communicate with a remote machine?
- Are there any free tools that can be used to communicate between two remote machines?
- If there are tools which can be used for remote commutation, then on what ports will they run to avoid any firewall issues?
- What sort of test bed environment is needed to test SIRAT?
- What sort of user interface is required for such tools?
- What are the reliability issues that one needs to be concerned for such tools?

- What sort of data dictionary might be needed and which technology needs to be used as a database?
- One of the major problems was how to execute a batch file on the remote machine and how to get the status once the batch file is executed.
- How to get the authentication system working?

8.3 Suggestions for Future Extensions to Project

The scope for this tool is very vast. This tool is mainly developed with a mind set to help Middleware Administrators. Most of the administrative capabilities are placed in SIRAT. But when suggestions are taken from other administrators like network or system administrators, then it is realized that there is a very high potential for improvement for this tool. Some of the major topics that SIRAT can be improved upon in the future are:

- **Authentication:** SIRAT does not have a proper authentication method in place. Currently, it allows anyone to register and use the tool. So an approval process to register and create an account is needed.

- **Authorization:** is not enabled in SIRAT. Currently, the project owners can see each other's projects, and can also execute the services of other projects. This will be a problem if the tool is in hosted mode.

- **Operating systems:** To enhance SIRAT to work for any operating system. Right now it is implemented only for Windows. With more operating systems, there will be more issues dealing with the reachability test page as well

- **Password Encryption:** Currently, the passwords that are stored in the database are not encrypted. So this can be improved in future.

- **Task scheduling:** Currently SIRAT does not have the functionality to execute a service at a scheduled time. It will be an added advantage if this feature is implemented.

There are other ideas that come to mind that are not so major in comparison to the items listed above. Some of these are

- Right now some customization on the remote batch file is needed so that it can output the result to a log file, and then be able to find out the status of the batch file by reading the log file. There might be another approach to improve upon this procedure.

- There might be other approaches to improve the overall design of the portal.

- Restart of remote machine facility could be implemented. This can be of help in situations such as updating installations or resetting boxes.

- With more experience in these kinds of projects, there is room for better coding of SIRAT.

Appendix A: Remote Administration

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

Description about Remote Administration

Remote administration refers to any method of controlling a computer from a remote location. Software that allows remote administration is becoming increasingly common and is often used when it is difficult or impractical to be physically near a system in order to use it, or in order to access web material that is not available in one's location, for example viewing the BBC iPlayer from outside the United Kingdom. A remote location may refer to a computer in the next room or one on the other side of the world. It may also refer to both legal and illegal (i.e. hacking) remote administration.

Any computer with an Internet connection, TCP/IP or on a Local Area Network can be remotely administered. For non-malicious administration, the user must install or enable server software on the host system in order to be viewed. Then the user/client can access the host system from another computer using the installed software. Usually, both systems should be connected to the internet, and the IP address of the host/server system must be known. Remote administration is therefore less practical if the host uses a dial-up modem, which is not constantly online and often has a Dynamic IP. Remote administration has developed and evolved over recent years, and is no longer simply about controlling a device over a network, being it local or wide area.

When looking to source a remote administration software solution, there are a number of more modern features that really make the difference. These features allow you, as an administrator, to save time, reduce energy costs and even track IT assets, by automating tasks, and administrating any of your IT resources, at any time, from any place.

Appendix B: Required Support Software to build SIRAT

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

As indicated, several software packages will be used to build SIRAT. This section will give a brief knowledge of those packages listed below:

- PsTools Suite
- JBoss
- JRockit
- MySQL

Appendix C: PsTools Suite

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

PsTools Suite is the most important set of tools that will be used to build SIRAT. These tools are used to communicate between the Remote Machines. PsTools Suite is developed by Sysinternals. **This software is a freeware but it cannot be distributed to others**. Consequently, if SIRAT is given to some vendor or a client, it is necessary to make sure this software is not bundled with SIRAT. PSTools Suite should be downloaded and installed separately at the location where SIRAT is installed.

The first tool in the suite was PsList, a tool that lets you view detailed information about processes, and the suite is continually growing. The "Ps" prefix in PsList relates to the Ps command in UNIX that provides a list of processes belonging to users.

PsTools Suite is a bundle of set of tools used for similar kind of tasks. This list of tools bundled in PsTools Suite is mentioned below;

- *PsExec* - execute processes remotely
- *PsFile* - shows files opened remotely
- *PsGetSid* - display the SID of a computer or a user
- *PsInfo* - list information about a system
- *PsKill* - kill processes by name or process ID
- *PsList* - list detailed information about processes
- *PsLoggedOn* - see who's logged on locally and via resource sharing (full source is included)
- *PsLogList* - dump event log records
- *PsPasswd* - changes account passwords
- *PsService* - view and control services
- *PsShutdown* - shuts down and optionally reboots a computer
- *PsSuspend* - suspends processes
- *PsUptime* - shows you how long a system has been running since its last reboot (PsUptime's functionality has been incorporated into *PsInfo*)

The tools all generally work in the same manner. On the host system, you launch the desired PsTool. Though many can be run locally, you'll find that these tools typically provide their utmost utility when run against a remote system or several systems at once. Using the command-line argument you provide, the utility (the tool itself) copies itself to the remote system's administrative share, which is the same as %SystemRoot% (C:\windows directory) on the remote system. Once copied into the remote directory, the application launches itself and installs a Windows service. This Windows service performs the task you requested, using the credentials you provided at launch time. If an end-user UI on the client system is required by the specific tool, this service then launches an additional binary in the user's context. The service communicates any necessary information back to the console that launched it. And finally, the service uninstalls itself.

In order to use the tools, you must have the following prerequisites: either Windows NT®, Windows 2000, Windows XP, Windows Vista, Windows 7 or Windows Server® 2003 (x86 or x64 versions of Windows are supported, but Itanium versions are not);

The Pre Conditions to make the PsTool Suite work

In order to make PsTools work, the below are pre conditions which should be implemented on the servers;

- **Ping** – The Remote machine should give positive response when you ping it. That means all the Remote Machines should in a reachable network
- **Net share** - ADMIN$ share should be enabled. Means all the Remote Machine C:\WINDOWS folder should be accessible.
- **Remote Registry service** - PsTools suite makes use of RPC calls via port 445. Make sure that the port 445 is opened in the firewall.
- **Administrator Rights** – Make sure that you have Administrator rights on the Remote Machines.

SIRAT does not use all the tools in suite. It mainly uses *PsExec, PsInfo* and *PsList*.

PsExec

PsExec is a light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software. PsExec's most powerful uses include launching interactive command-prompts on remote systems and remote-enabling tools like IpConfig that otherwise do not have the ability to show information about remote systems.

Here is a simplified usage for PsExec:

PsExec \\computer -u username -p password command

PsExec switches

Switch	Description
-a	Specifies processor affinity.
-c	Indicates whether the application should be copied to the remote system.
-d	Specifies not to wait for the process to finish before exiting (it won't wait for a success or failure message).
-e	Loads using the specified account's profile.
-f	Copies the application even if it exists on the remote system. Ensures that you are running the current (or a specific) version.
-i	Runs the application interactively on the remote system (required if you want an end user to see any user interface from the application).
-l	Runs the application as a limited user. Removes the Administrators group token and strips process privileges to the level they would be for a member of the Users group. Handy to run high-risk Internet-facing applications with minimal privileges.
-n	Specifies the timeout to wait before failing to connect to remote computers.
-priority	Runs the process at the specified priority. That is, specifies how Windows should set the process priority for a given task (-low, -below normal, -above normal, -high or -realtime).
-s	Runs the process using the System account (handy to run Regedit to access privileged registry keys).
-v	Copies the application only if the one being copied is newer.
-x	Displays UI on the Winlogon secure desktop-only works on the local system.
@file	Runs the command on all computers specified in the text file.

PsInfo

PsInfo is used to provide detailed system information. This obviously would be the most useful when scripted against multiple remote systems, with the output sent to a file (such as an Excel® and database-friendly CSV file). Note that this utility operates via an important concept that is different from most of the other tools: if you don't provide a computer name, PsInfo will, by default, attempt to query all computers in the domain.

PsInfo's usage is as follows:

Copy Code
PsInfo \\computer -u username -p password

While the syntax is simple, additional switches and parameters are also available as shown below;

PsInfo Switches

Switch	Description
-h	Shows all Windows updates.
-s	Shows all applications installed on the system.
-d	Shows disk/volume information for the system.
-c	Outputs the results in CSV format.
-t	Specifies another delimiter for the CSV file other than the default value of \t.
Filter	Shows only data that matches a given filter.
@file	Runs the command on all computers listed in the specified text file.

PsList

PsList is able to capture the list of one or more processes running on a remote system.

Usage for PsList is pretty simple:

Copy Code
PsList \\computer -u username -p password account

As with the other tools, the account parameter can specify a user, computer, or group. Additional parameters are shown below;

PsList Switches

Switch	Description
-t	Shows information at the thread level.
-m	Provides detailed memory information for each process.
-x	Shows process, memory, and thread information.
-t	Shows information as a process tree (think Process Explorer tree view).
-s [n]	Runs in task manager mode for the specified number of seconds. This allows for a relatively real-time display of remote process information.

-r [n]	Specifies how frequently (in seconds) to refresh when running in task manager mode.
Name	Shows information about processes that begin with the name specified.
-e	Matches the process name exactly.
Pid	Displays information about a specific process.

Appendix D: JBoss Application Server

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

SIRAT is built on J2EE technology. JBoss will be used to host the SIRAT code.

JBoss Application Server is a free software/open-source Java EE-based application server. An important distinction for this class of software is that it not only implements a server that runs on Java, but it actually implements the Java EE part of Java. Because it is Java-based, the JBoss application server operates cross-platform: usable on any operating system that supports Java. JBoss Application Server was developed by JBoss, now a division of Red Hat.

Product Features

Aspect-Oriented Programming (AOP) support
Clustering
Deployment API
Distributed caching (using JBoss Cache, a standalone product)
Distributed deployment (farming)
Enterprise Java Beans versions 3 and 2.1
Failover (including sessions)
Hibernate-integration (for persistence programming; JPA)
JAAS (Java Authentication and Authorization Service)
JCA (Java Connector Architecture)-integration
Java Management Extensions
JMS (Java Message Service) integration
JNDI (Java Naming and Directory Interface)
JTA (Java Transaction API)
JACC (Java Authorization Contract for Containers)-integration
JavaMail
JavaServer Faces 1.2 (Mojarra)

JSP/Servlet 2.1/2.5 (Tomcat)
JBossWS (JBoss Web Services) for Java EE-Web Services like JAX-WS
JDBC
Load balancing
Management API
OSGi framework
RMI-IIOP (JacORB, contraction of Java and CORBA)
SAAJ (SOAP with Attachments API for Java)
Teiid data virtualization system
4.5.2.2 Licensing and Pricing

JBoss itself is open source, but Red Hat charges to provide a support subscription for JBoss Enterprise Middleware. Before November 2010 JBoss was licensed as annual subscription in bundles of 4 and 32 CPU sockets. As of November 2010 the licensing changed and all cores on the system are now counted. The new core bundles licensing is now available for 16 and 64 cores. The JBoss Enterprise Application Platform (EAP) with management option 16 core bundle premium support costs $11,250 per year. Complete JBoss pricing is available.

Appendix E: JRockit JVM

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

In this project we have deiced to use JRockit as JVM for the JBoss application server. This will be a separate installation and will need to be configured in JBoss. JRockit, a proprietary Java Virtual Machine (JVM) originally developed by Appeal Virtual Machines and acquired by BEA Systems in 2002, became part of Oracle Fusion Middleware in 2008.

The JRockit code base and the HotSpot virtual machine from Sun Microsystems (now Oracle) are currently being integrated, with the target of releasing a JVM with a combined code base around the release date of JDK 8. JRockit was made free and publicly available in May 2011. Many JRE class files distributed with JRockit exactly replicate those distributed by Sun. JRockit overrides class files which relate closely to the JVM, therefore retaining API compatibility while enhancing the performance of the JVM. The JRockit JVM is a high performance JVM developed to ensure reliability, scalability, manageability, and flexibility for Java applications. The JRockit JVM delivers a new level of performance for Java applications deployed on Intel 32-bit (Xeon) and 64-bit (Xeon, Itanium, and SPARC) architectures at significantly lower costs to the enterprise. Furthermore, it is the only enterprise-class JVM optimized for Intel architectures,

providing seamless inter operability across multiple hardware and operating configurations. The JRockit JVM makes it possible to gain optimal performance for your Java applications when running it on either the Windows or Linux operating platforms on either 32-bit or 64-bit architectures. The JRockit JVM is especially well suited for running Oracle WebLogic Server.

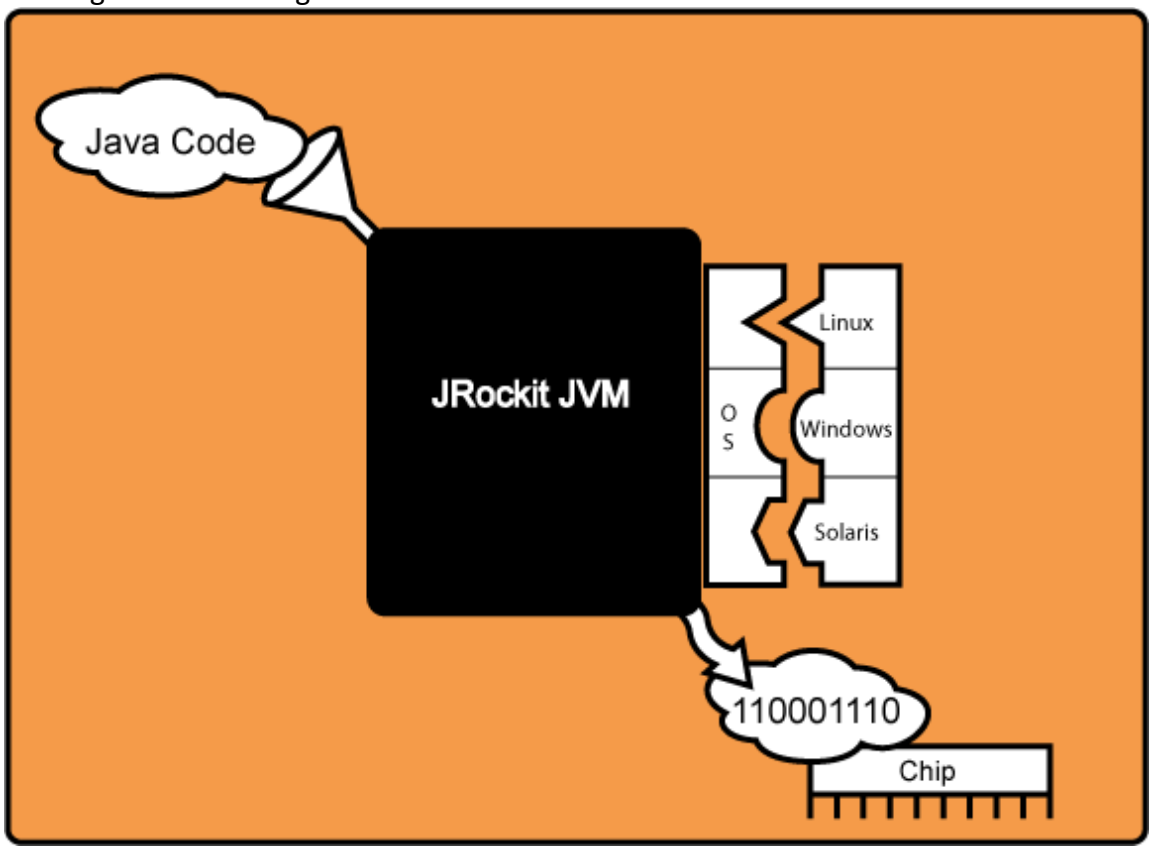


Figure 10 JRockit JVM as a Black Box

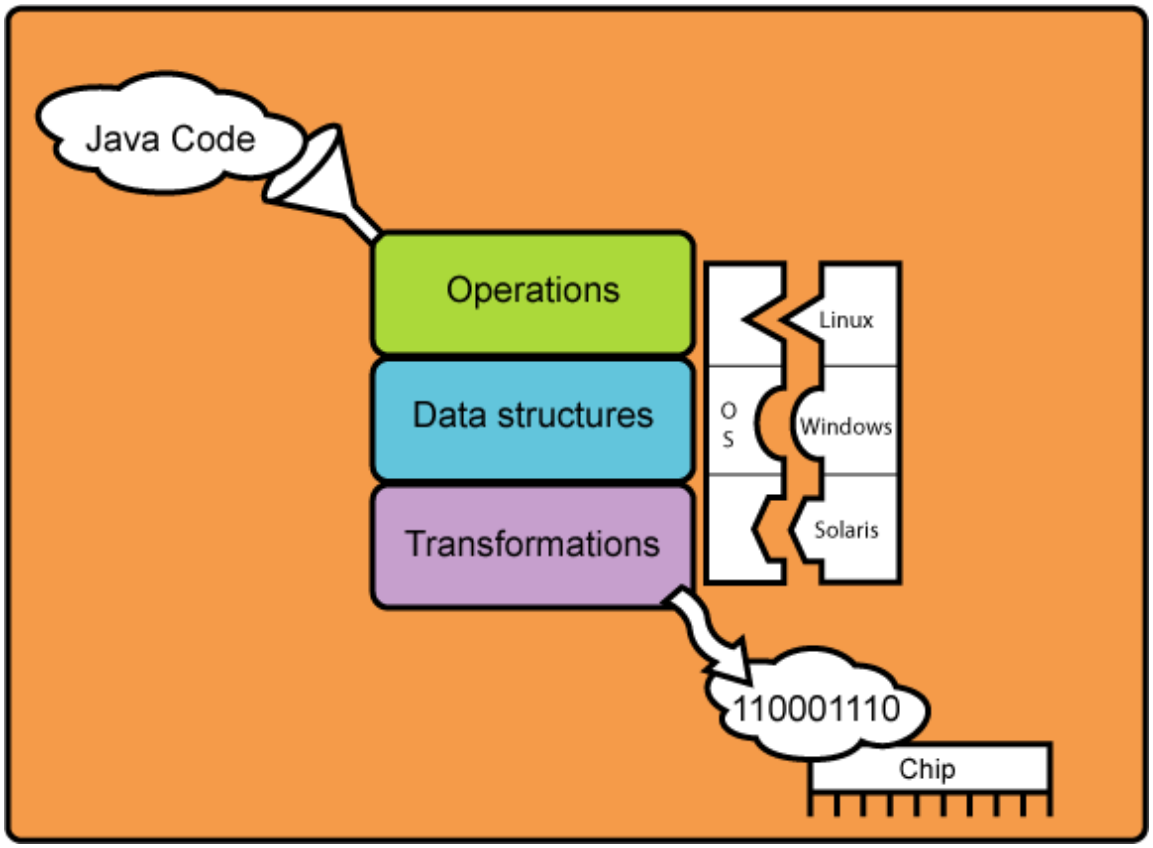


Figure 11 Java Code Compilations

How the JRockit JVM Compiles Code

The code generator in the JRockit JVM runs in the background during the entire run of your Java application, automatically adapting the code to run its best. The code generator works in three steps, as described below

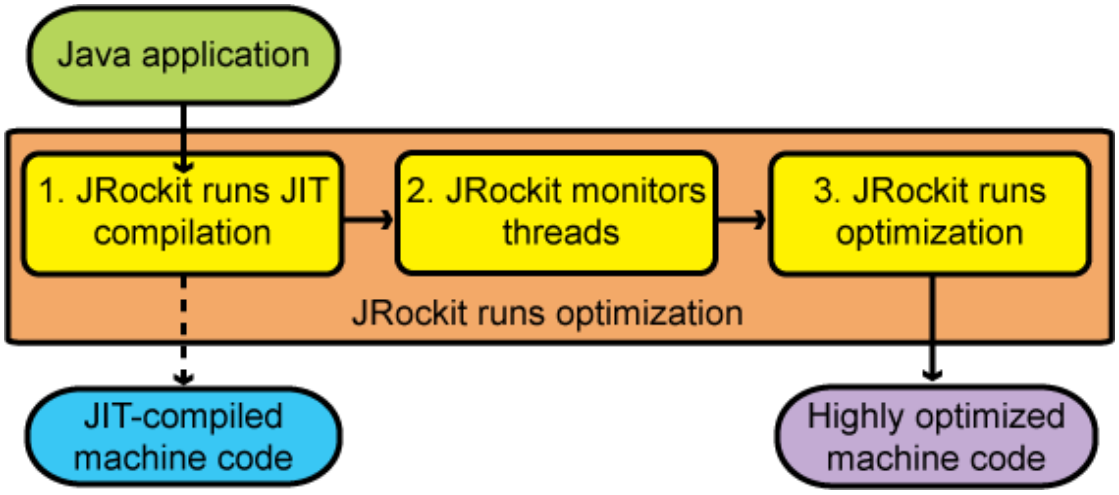


Figure 12 How the JRockit JVM Optimizes Code for Your Java Application

Appendix F: MySQL Database

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

This project uses MySQL as the backend database. This tool does not use the database intensively but used to store the information of Remote Machines, like names of the machines, username's and passwords etc. MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com. The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.
MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in. MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's.

The MySQL database has become the world's most popular open source database because of its high performance, high reliability and ease of use. It is also the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) Many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, Alcatel Lucent and Zappos rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software.

MySQL runs on more than 20 platforms including Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX, giving you the kind of flexibility that puts you in control. Whether you're new to database technology or an experienced developer or DBA, MySQL offers a

comprehensive range of database tools, support, training and consulting services to make you successful.

Appendix G: Oracle Weblogic Application Server

Note: This section is an amalgamation from various websites referenced in the Reference section. All the material discussed here will help us better understand the concepts used in building SIRAT.

We will be using this Middleware Application Server to integrate the Remote Admin Tool. We will attempt to use this Admin Tool to do all the administrative task of this application server. Weblogic is owned by Oracle Corporation; Oracle WebLogic consists of a Java EE platform product-family that includes:

A Java EE application server, WebLogic Application Server
An enterprise portal, WebLogic Portal
An Enterprise Application Integration platform
A transaction server and infrastructure, WebLogic Tuxedo
A telecommunication platform, WebLogic Communication Platform
An HTTP web server

WebLogic is a server software application that runs on a middle tier, between back-end databases and related applications and browser-based thin clients. WebLogic is a leading e-commerce online transaction processing (OLTP) platform, developed to connect users in a distributed computing environment and to facilitate the integration of mainframe applications with distributed corporate data and applications. WebLogic server is based on Java 2 Platform, Enterprise Edition (J2EE), the standard platform used to create Java-based multi-tier enterprise applications. J2EE platform technologies were developed through the efforts of BEA Systems and other vendors in collaboration with the main developer, Sun Microsystems. Because J2EE applications are standardized modules, WebLogic can automate many system-level tasks that would otherwise have demanded programming time. The main features of WebLogic server include connectors that make it possible for any legacy application on any client to interoperate with server applications, Enterprise Java Bean (EJB) components, resource pooling, and connection sharing that make applications very scalable. An administration console with a user interface makes management tasks more efficient and features such as Secure Sockets Layer (SSL) support for the encryption of data transmissions, as well as authentication and authorization mechanisms make applications and transactions secure.

The diagram below will give a high level understanding about the Weblogic domain structure. In a domain you have Admin Server and Managed Servers as main

components. There will be several other components which will be on the managed servers. The Admin Servers is mainly responsible for the control of these components. Like start/stop and other administrative work. In this project we will try to integrate our tool to the administration of Weblogic Domain.

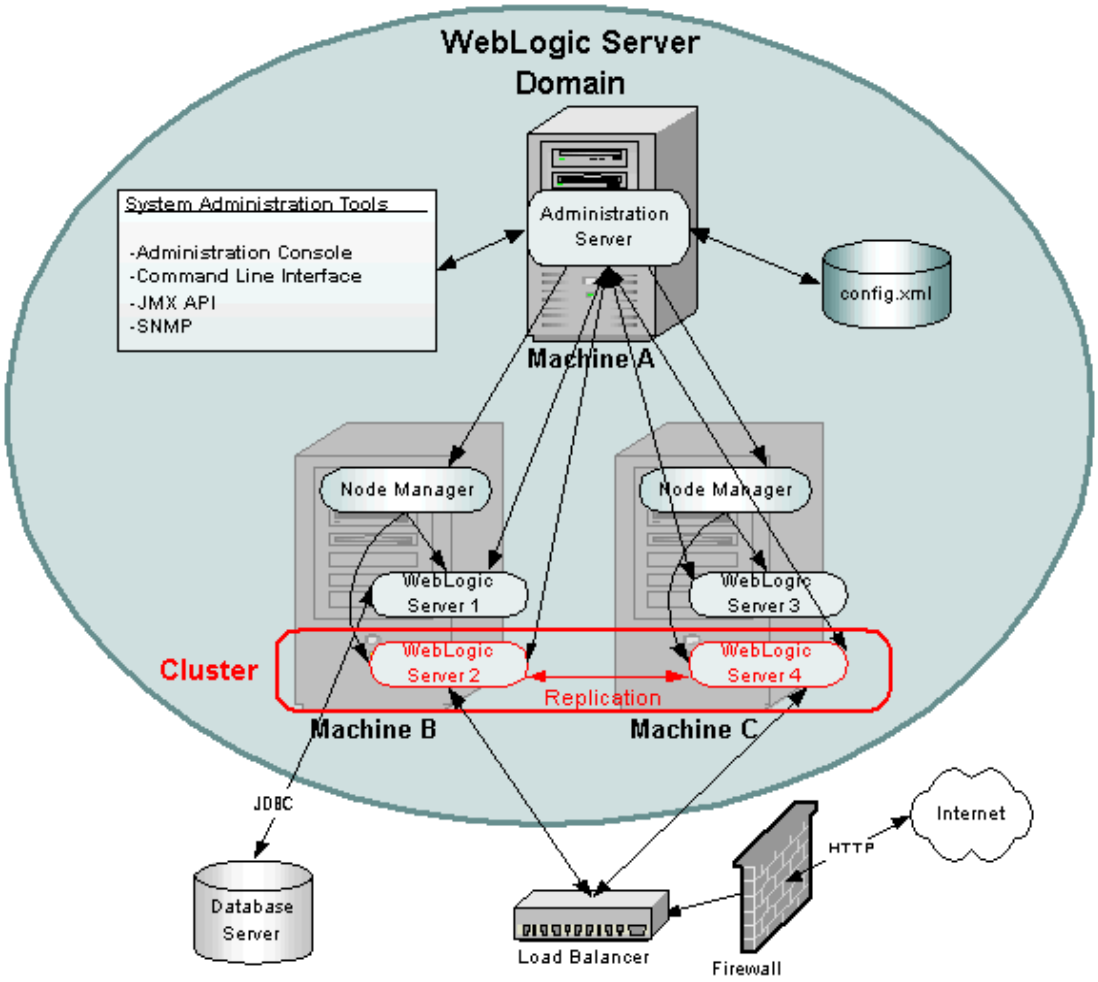


Figure 13 The Weblogic Server life cycle is shown in diagram below

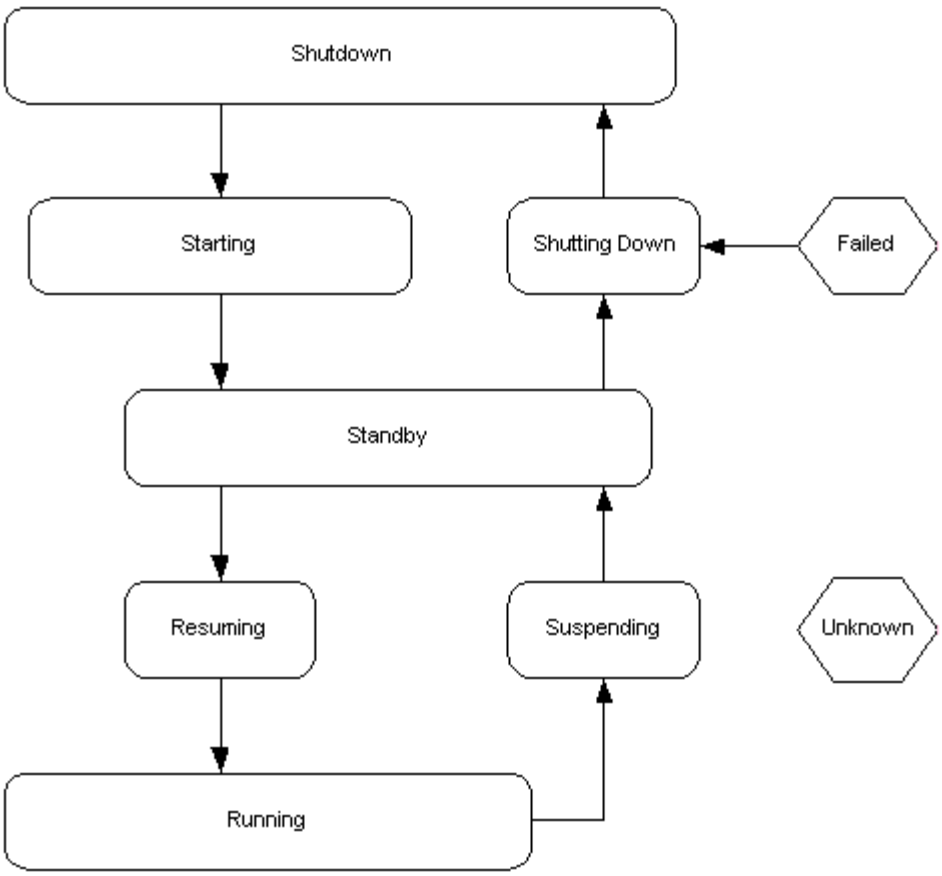


Figure 14 Weblogic Server life cycle

Appendix H: MySQL 5.5.9 Installation

To run MySQL on Windows, the following is needed:

A Windows operating system such as Windows 2000, Windows XP, Windows Vista, Windows 7, Windows Server 2003, or Windows Server 2008. Both 32-bit and 64-bit versions are supported. Generally, MySQL is installed using an account that has administrator rights. Otherwise, some problems might be encountered such as editing the PATH environment variable or accessing the *Service Control Manager*. Once installed, MySQL does not need to be executed using a user with Administrator privileges.

MySQL for Windows is available in several distribution formats:

Binary distributions are available that contain a setup program that installs everything needed to start the server immediately. Another binary distribution format contains an archive that can be unpacked in the installation location and then configured.

After installing MySQL Server, it is recommended to disable virus scanning on the main directory (datadir) being used to store the MySQL table data. There is usually a system built into the virus scanning software to enable certain directories to be specifically ignored during virus scanning.

Another alternative is the MSI package designed to install and configure MySQL quickly.

The MySQL Installation Wizard and MySQL Configuration Wizard are available in the complete install package of MySQL 5.5.9.

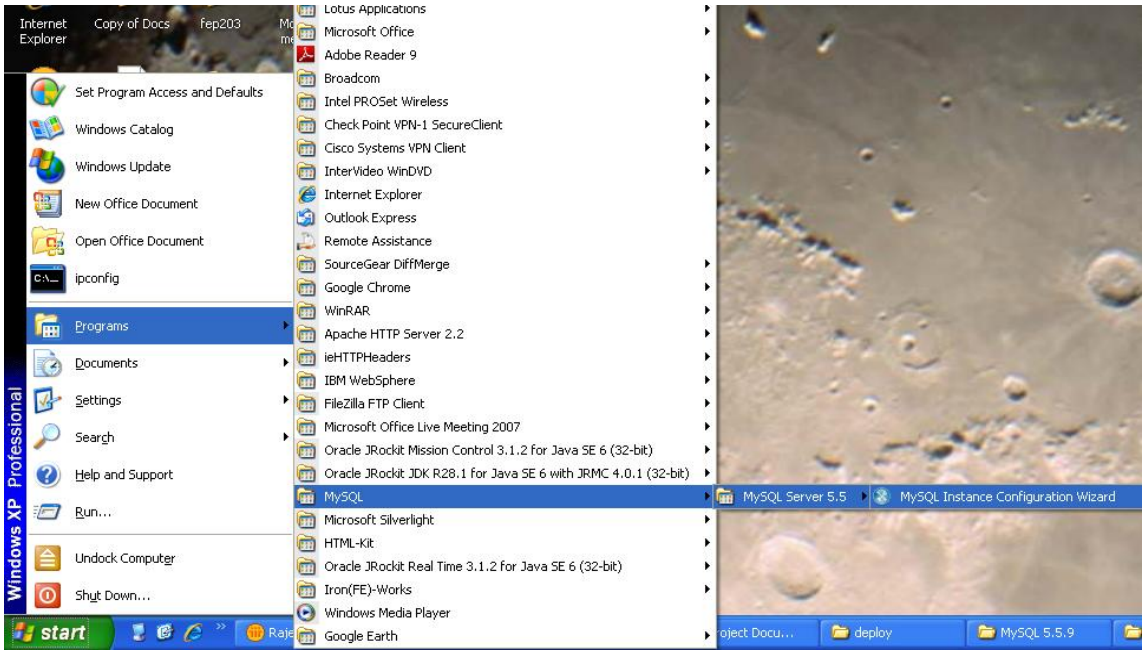
Note: All the installable files are placed in C:\RemoteAdmin\Installables of SIRAT or it can be downloaded from the corresponding websites. Please keep this in mind for the rest of the document.

You can download the installation file from the website or you can use the file kept in C:\RemoteAdmin\Installables\MySQL 5.5.9 folder.

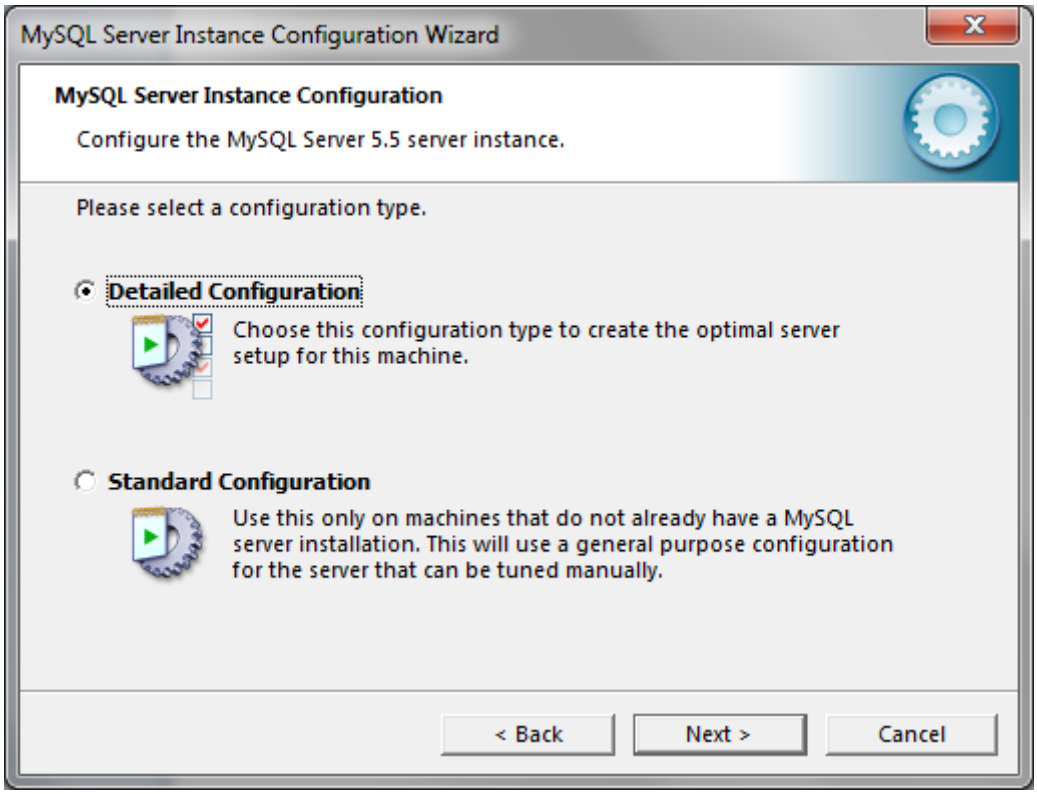
The MySQL install file has .msi extention (MSI is similar to EXE extention). Installing the .msi file is simple. Double click the file and go through the screens.

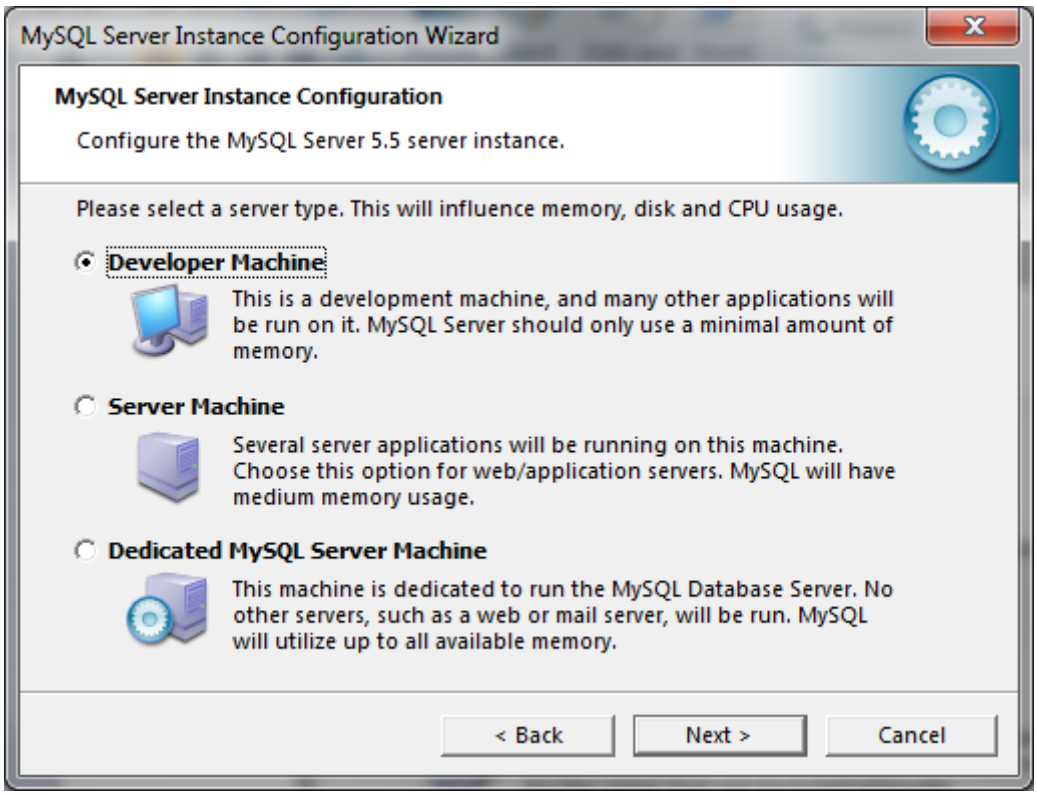
During installation the name of the windows service is provided. It can be what ever you want. In SIRAT case the name is given as MySQL. Once installation is complete, now it is required to start the configuration wizard and configure the database.

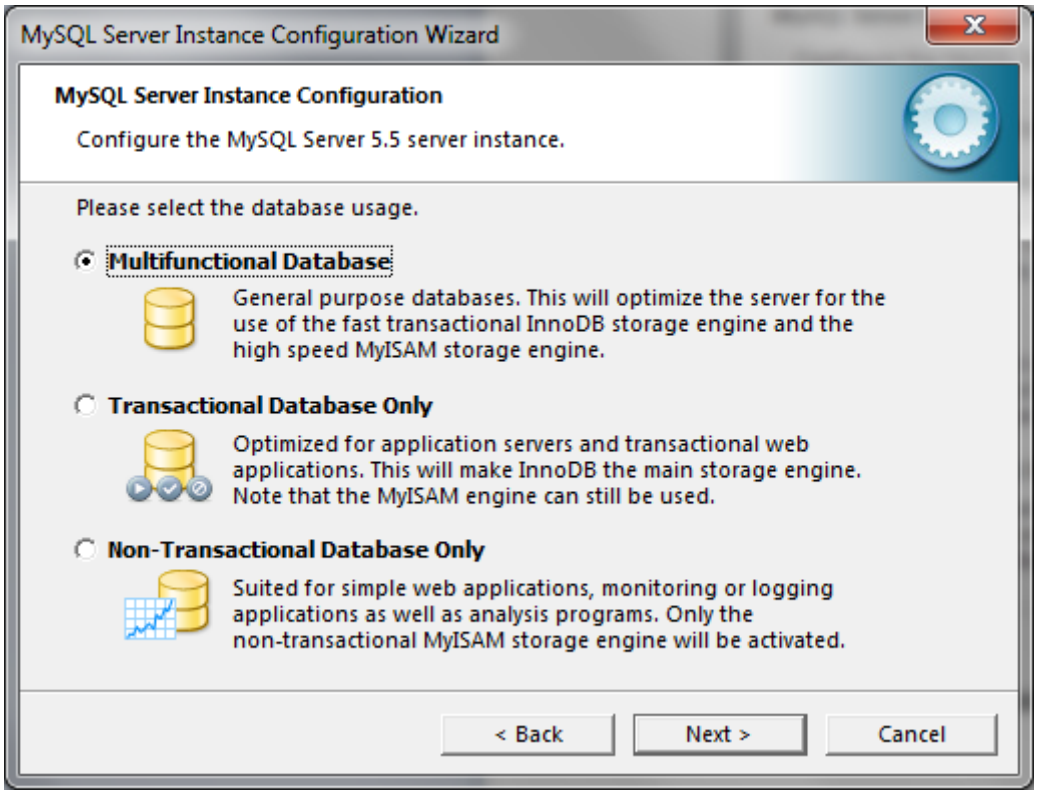
Start the configuration wizard by going into program files through start button.

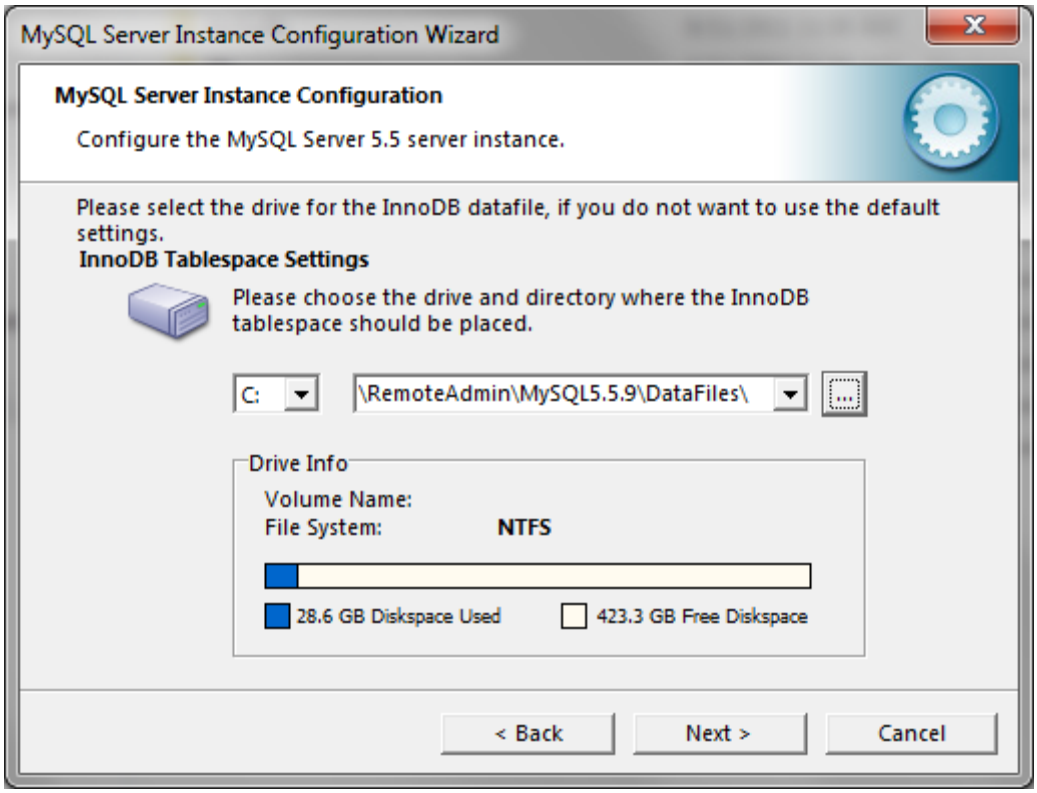


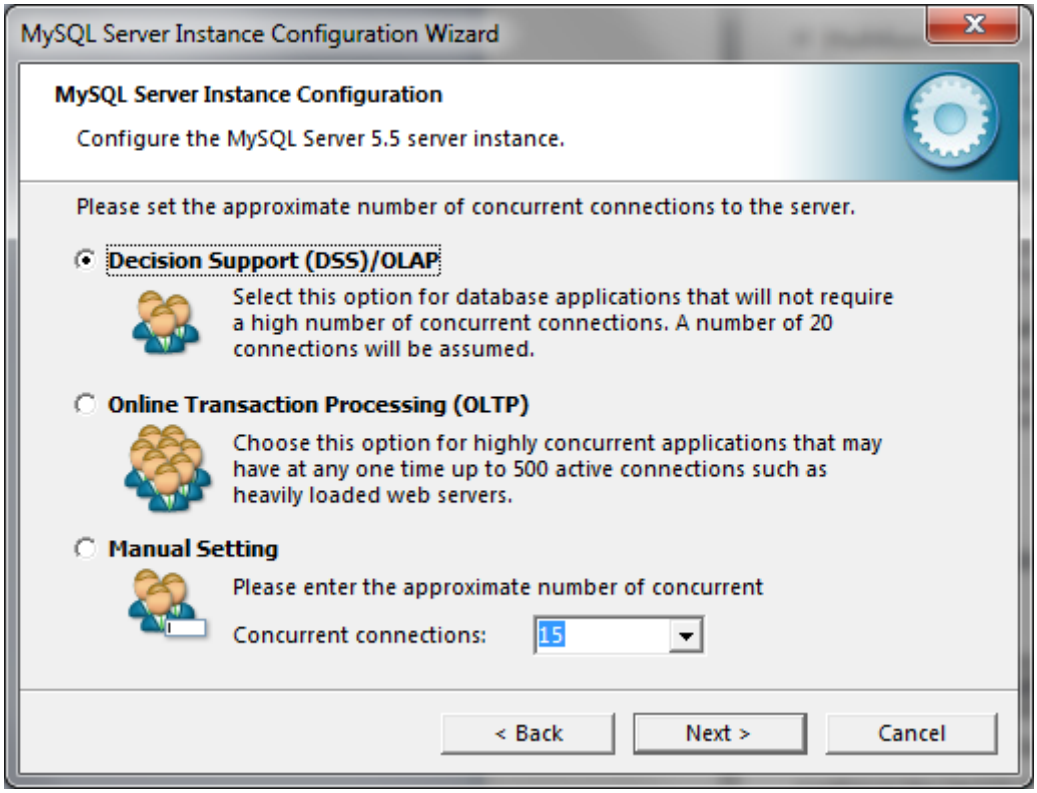
It will open the MySQL configuration wizard. Follow the screenshots below;

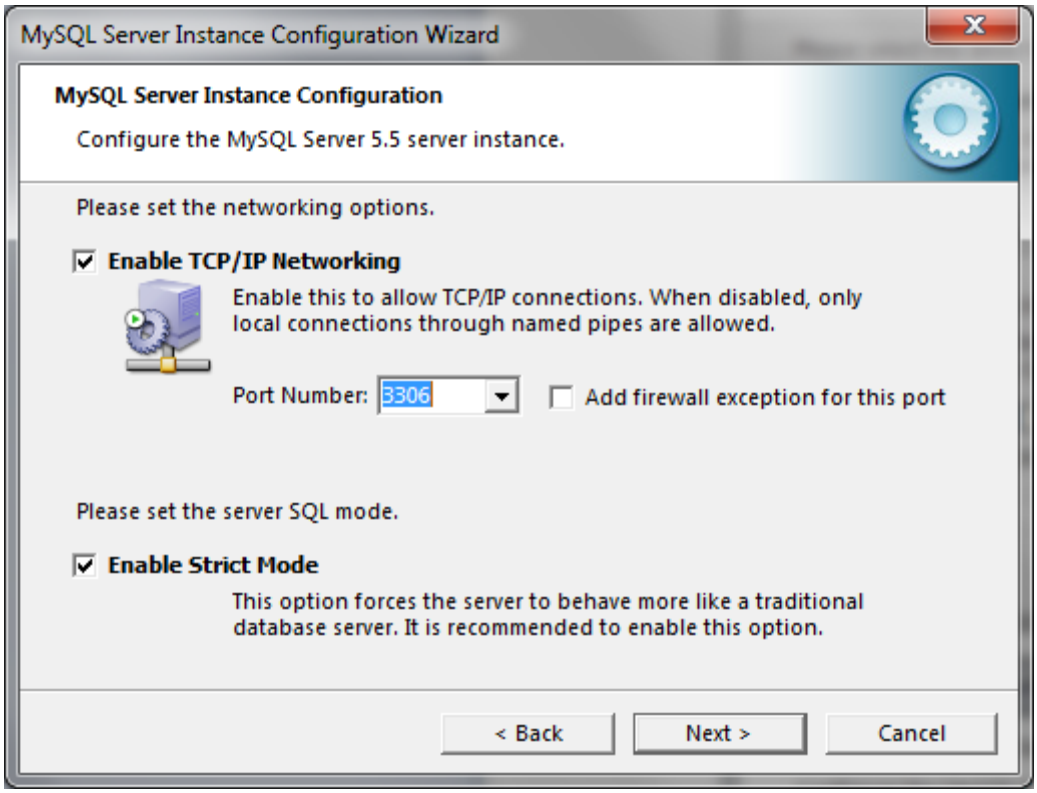

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please select a configuration type.
Detailed Configuration
Choose this configuration type to create the optimal server setup for this machine.
Standard Configuration
Use this only on machines that do not already have a MySQL server installation. This will use a general purpose configuration for the server that can be tuned manually.
< Back
Next >
Cancel

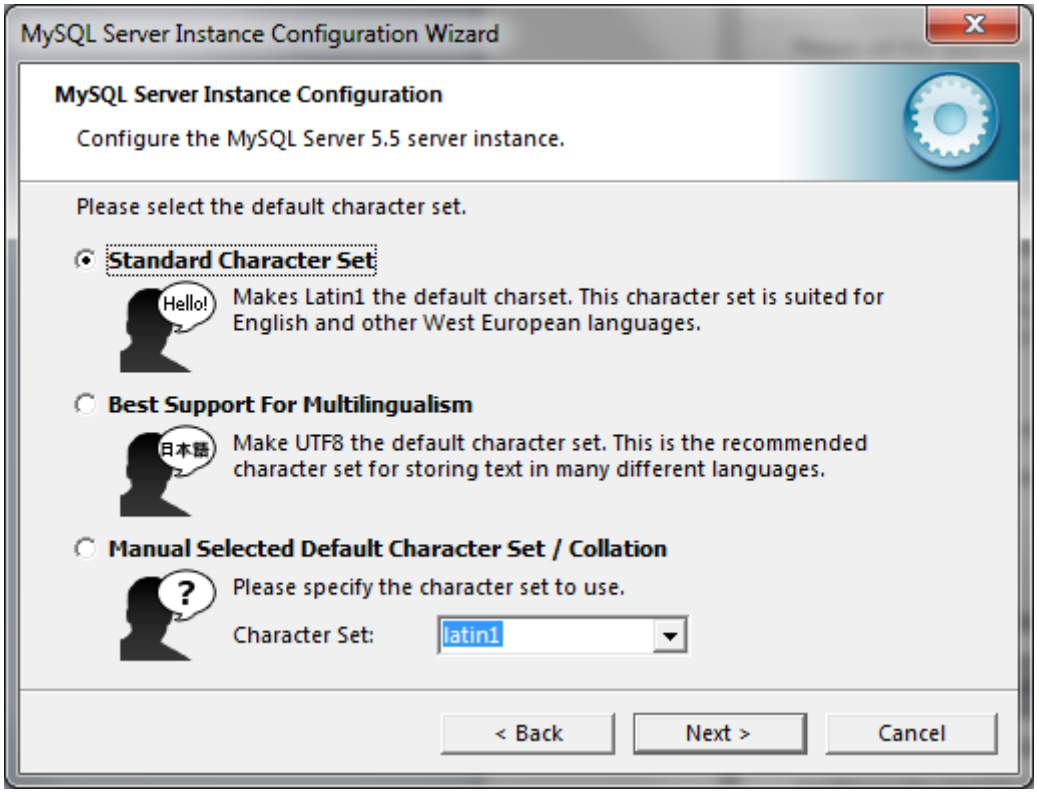

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please select a server type. This will influence memory, disk and CPU usage.
Developer Machine
This is a development machine, and many other applications will be run on it. MySQL Server should only use a minimal amount of memory.
Server Machine
Several server applications will be running on this machine. Choose this option for web/application servers. MySQL will have medium memory usage.
Dedicated MySQL Server Machine
This machine is dedicated to run the MySQL Database Server. No other servers, such as a web or mail server, will be run. MySQL will utilize up to all available memory.
< Back
Next >
Cancel

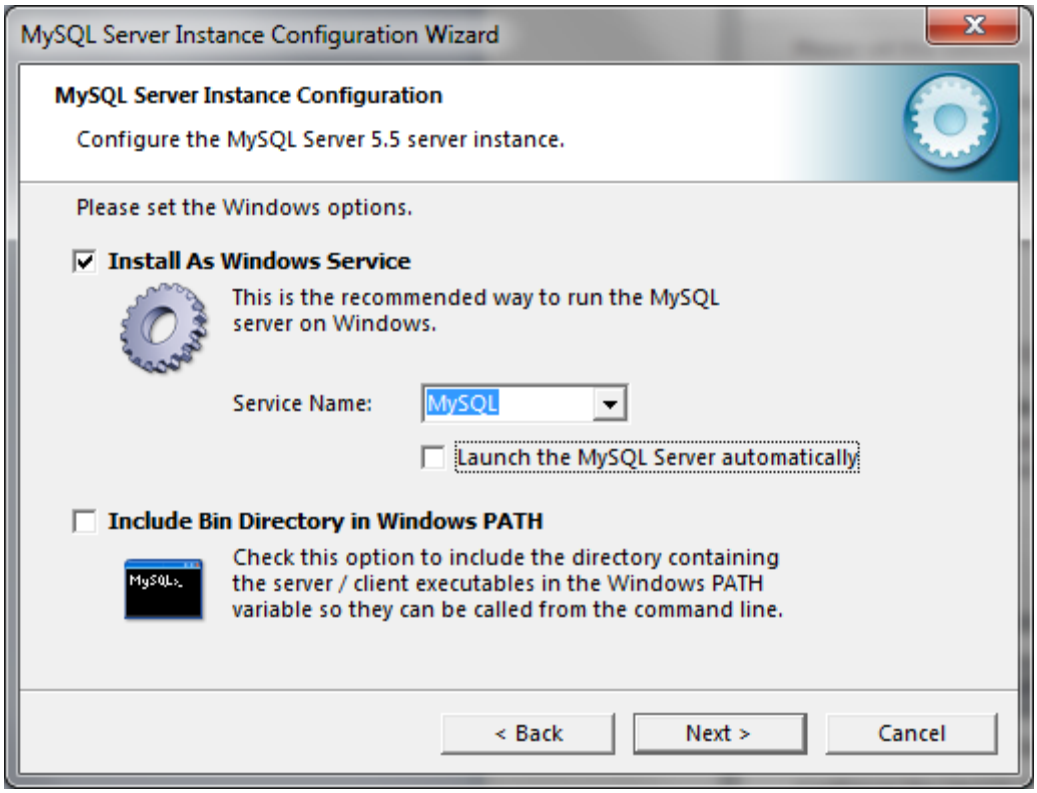

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please select the database usage.
Multifunctional Database
General purpose databases. This will optimize the server for the use of the fast transactional InnoDB storage engine and the high speed MyISAM storage engine.
Transactional Database Only
Optimized for application servers and transactional web applications. This will make InnoDB the main storage engine. Note that the MyISAM engine can still be used.
Non-Transactional Database Only
Suited for simple web applications, monitoring or logging applications as well as analysis programs. Only the non-transactional MyISAM storage engine will be activated.
< Back
Next >
Cancel

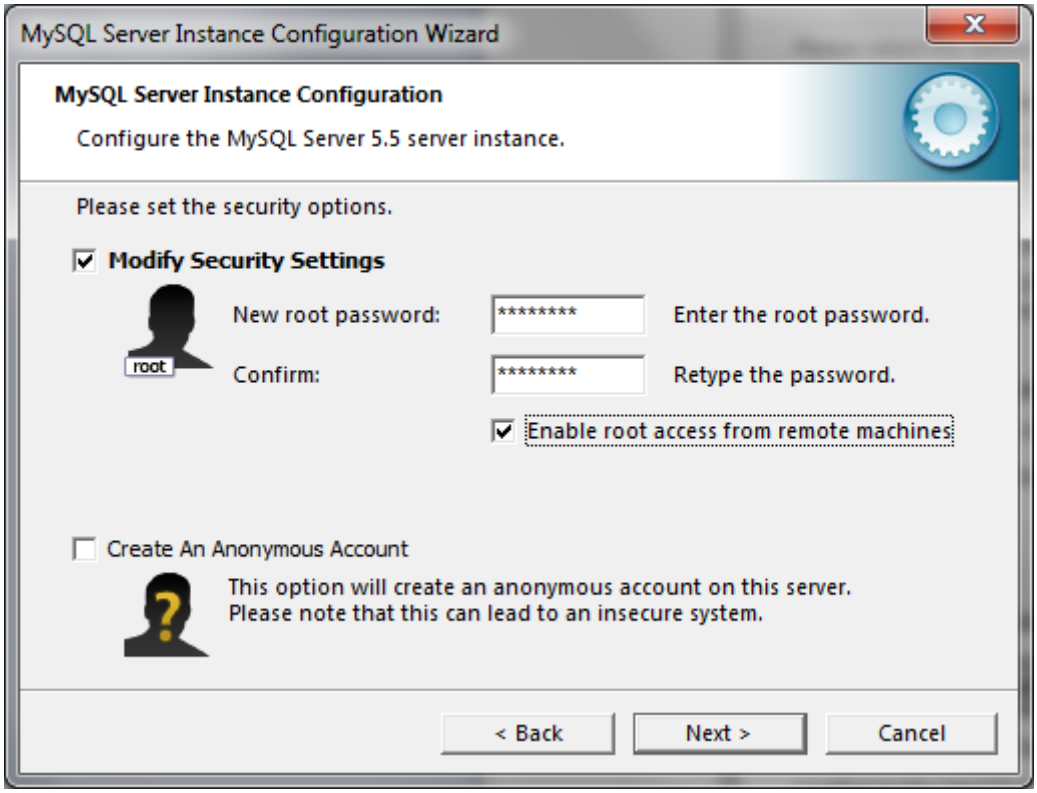

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please select the drive for the InnoDB datafile, if you do not want to use the default settings.
InnoDB Tablespace Settings
Please choose the drive and directory where the InnoDB tablespace should be placed.
C:
\RemoteAdmin\MySQL5.5.9\DataFiles\
...
Drive Info
Volume Name:
File System: NTFS
28.6 GB Diskspace Used
423.3 GB Free Diskspace
< Back
Next >
Cancel

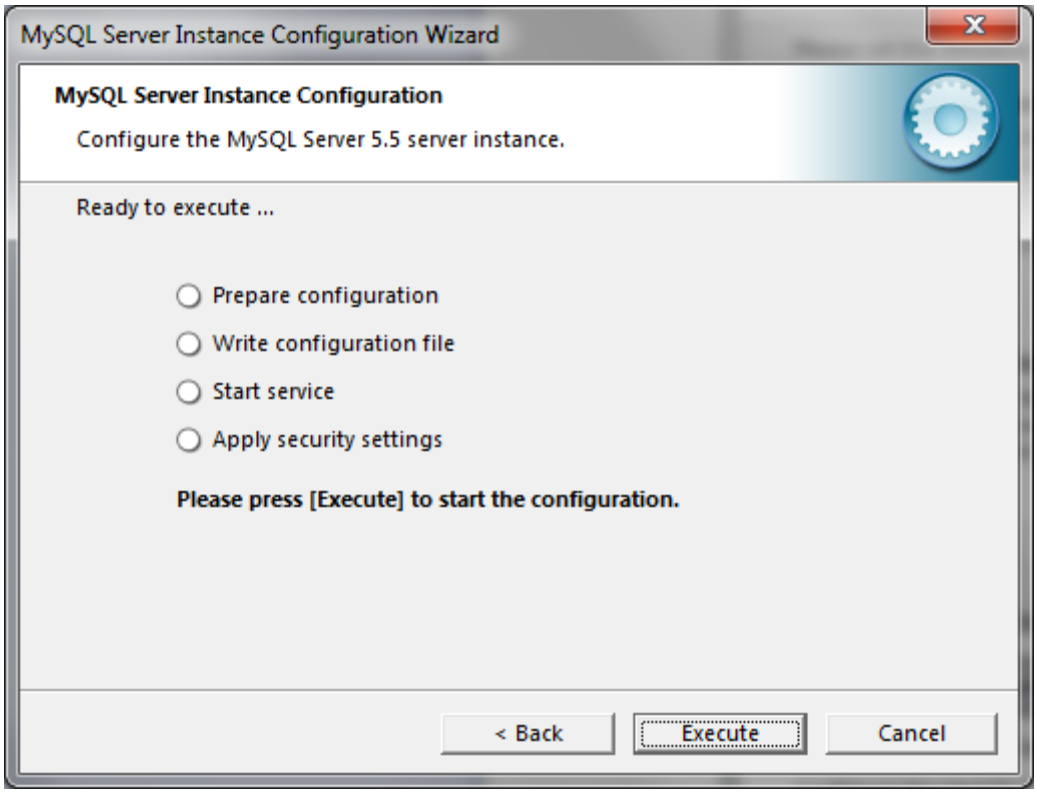

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please set the approximate number of concurrent connections to the server.
Decision Support (DSS)/OLAP
Select this option for database applications that will not require a high number of concurrent connections. A number of 20 connections will be assumed.
Online Transaction Processing (OLTP)
Choose this option for highly concurrent applications that may have at any one time up to 500 active connections such as heavily loaded web servers.
Manual Setting
Please enter the approximate number of concurrent
Concurrent connections: 15
< Back
Next >
Cancel


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please set the networking options.
Enable TCP/IP Networking
Enable this to allow TCP/IP connections. When disabled, only local connections through named pipes are allowed.
Port Number: 3306
Add firewall exception for this port
Please set the server SQL mode.
Enable Strict Mode
This option forces the server to behave more like a traditional database server. It is recommended to enable this option.
< Back
Next >
Cancel

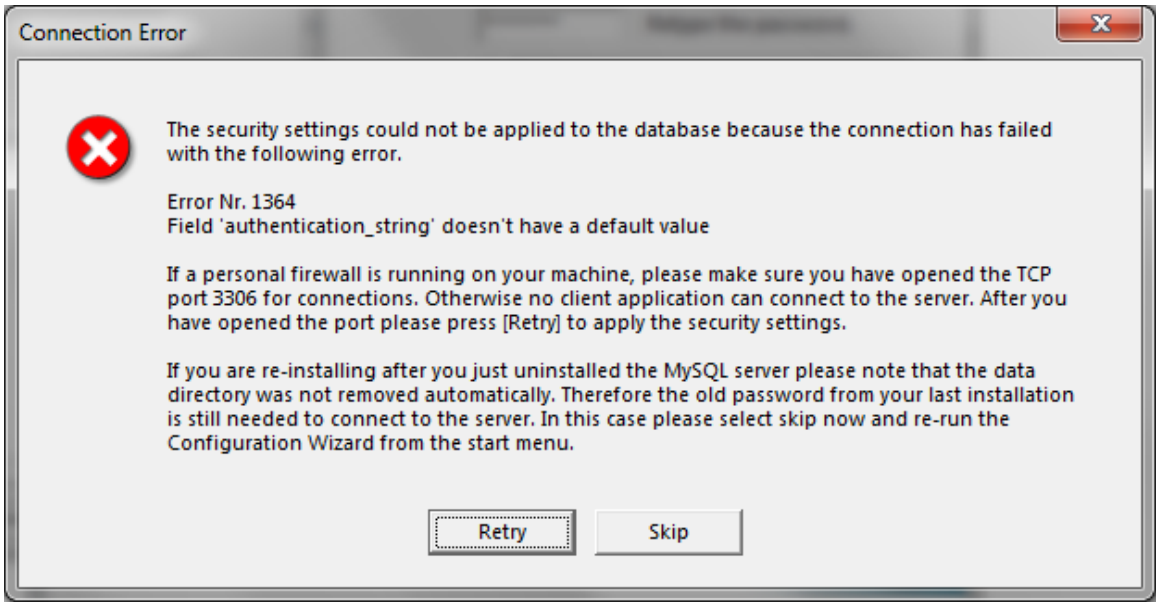

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please select the default character set.
Standard Character Set
Hello!
Makes Latin1 the default charset. This character set is suited for English and other West European languages.
Best Support For Multilingualism
日本語
Make UTF8 the default character set. This is the recommended character set for storing text in many different languages.
Manual Selected Default Character Set / Collation
?
Please specify the character set to use.
Character Set: latin1
< Back
Next >
Cancel

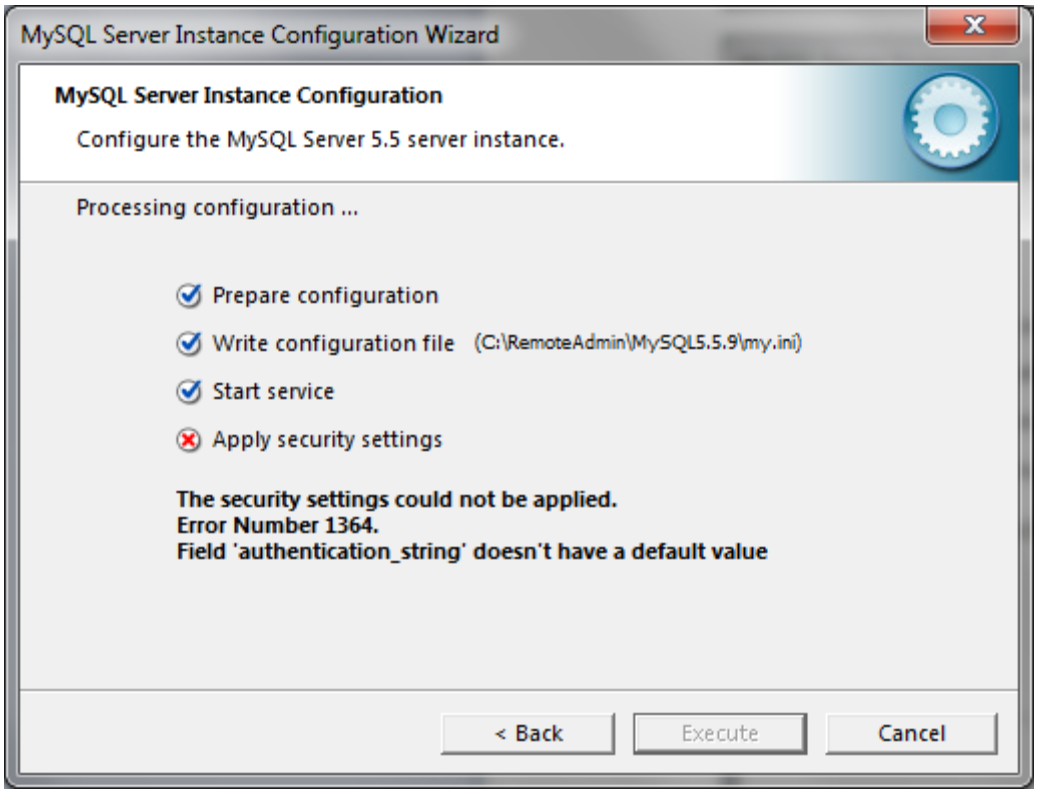

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please set the Windows options.
Install As Windows Service
This is the recommended way to run the MySQL server on Windows.
Service Name:
MySQL
Launch the MySQL Server automatically
Include Bin Directory in Windows PATH
Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.
< Back
Next >
Cancel


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.5 server instance.
Please set the security options.
Modify Security Settings
root
New root password:
****
Enter the root password.
Confirm:
****
Retype the password.
Enable root access from remote machines
Create An Anonymous Account
This option will create an anonymous account on this server.
Please note that this can lead to an insecure system.
< Back
Next >
Cancel



Note: You can avoid the security settings to avoid below Connection Error. Even though it fails with below error, the windows service is still created and the installation works.


Connection Error
The security settings could not be applied to the database because the connection has failed with the following error.
Error Nr. 1364
Field 'authentication_string' doesn't have a default value
If a personal firewall is running on your machine, please make sure you have opened the TCP port 3306 for connections. Otherwise no client application can connect to the server. After you have opened the port please press [Retry] to apply the security settings.
If you are re-installing after you just uninstalled the MySQL server please note that the data directory was not removed automatically. Therefore the old password from your last installation is still needed to connect to the server. In this case please select skip now and re-run the Configuration Wizard from the start menu.
Retry
Skip



Please note, as specified in the note, even though it failed at security settings, the installation works fine and you can close the screen to complete the process.

Note: Microsoft Windows XP and later include a firewall which specifically blocks ports. If you plan on using MySQL through a network port then you should open and create an exception for this port before performing the installation.

To check and if necessary add an exception to the firewall settings;

First ensure that you are logged in as an Administrator or a user with Administrator privileges.

- Go to the Control Panel, and double click the Windows Firewall icon.
- Choose the Allow a program through Windows Firewall option and click the Add port button.
- Enter MySQL into the Name text box and **3306** (or the port of your choice) into the Port number text box.
- Also ensure that the TCP protocol radio button is selected. If you wish, you can also limit access to the MySQL server by choosing the Change scope button.
- Confirm your choices by clicking the OK button.

Additionally, when running the MySQL Installation Wizard on Windows Vista or newer, ensure that you are logged in as a user with administrative rights.

Appendix I: Installing the JRockit 1.6.0_20

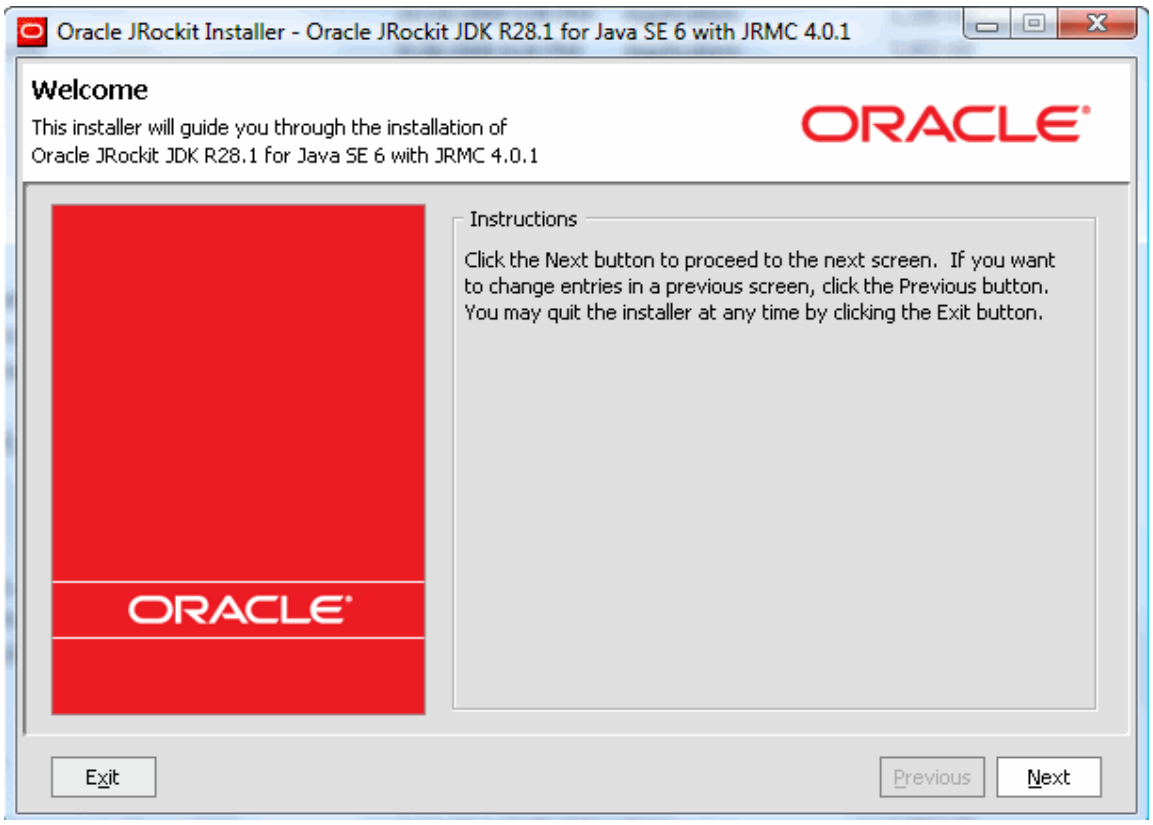
You can download the JRockit version used while building SIRAT or the latest version from the Oracle JRockit page. Get the correct version with respect the OS you are using;

Note: JDK for the JVM should be decided while developing the Java code. On whatever JDK the Java code is developed, we need to use that particular JVM for the application Server.

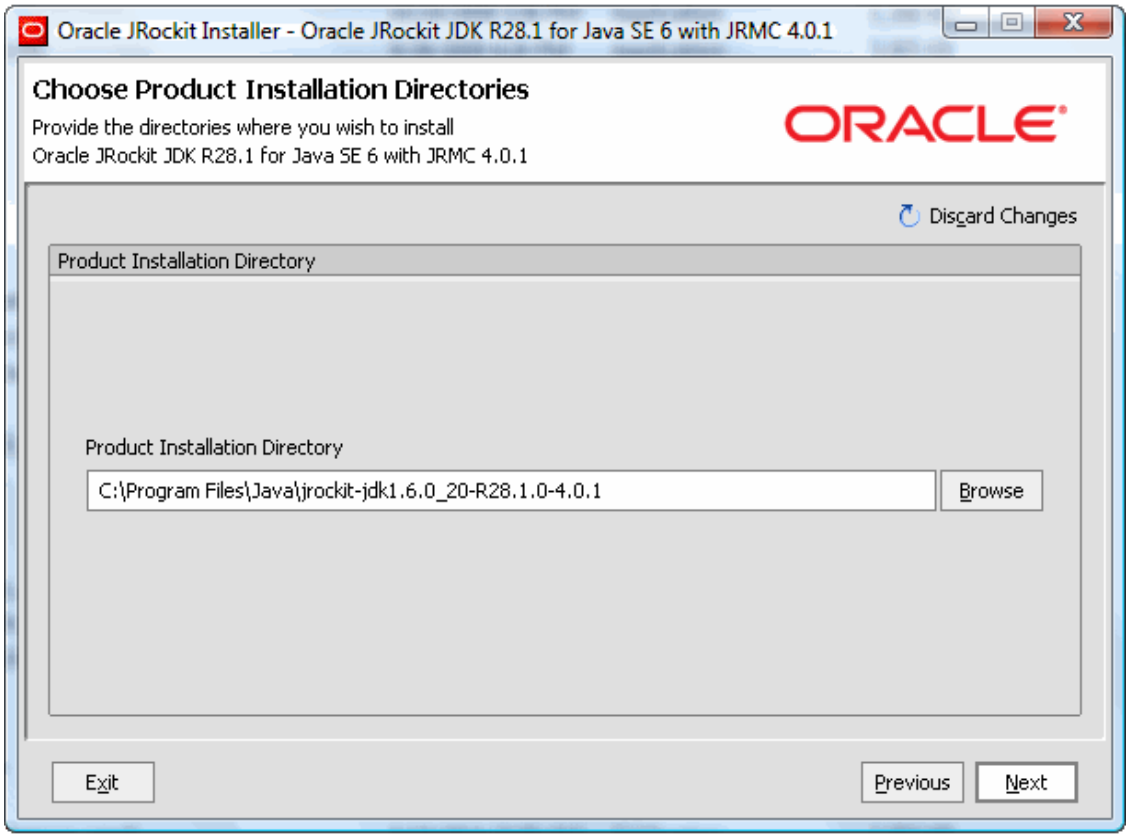
Platform...	Use...
Windows 32-bit x86	*product_name_version*-windows-ia32.exe
Windows 64-bit Xeon/AMD64	*product_name_version*-windows-x64.exe

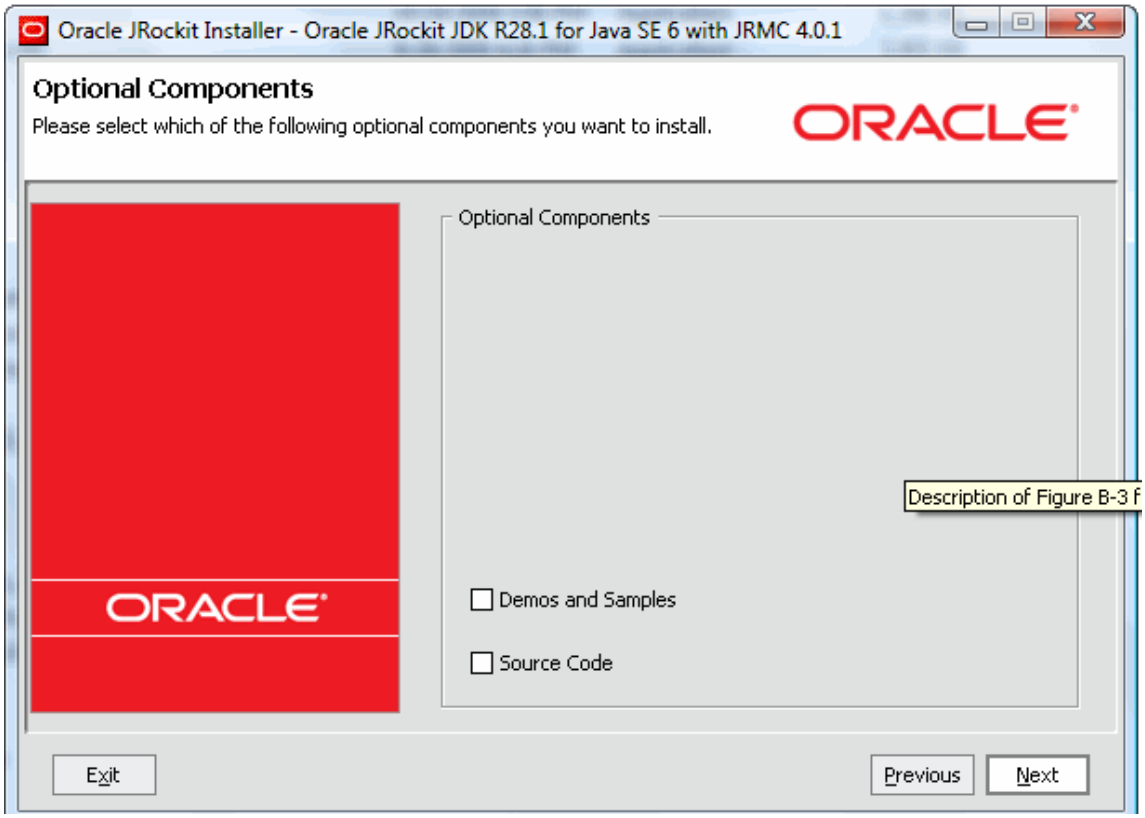
Download the appropriate file and install in C:\RemoteAdmin\Jrockit<version name>

On your local drive, locate the installer for the version of the product you want to install. Run the installer and follow the screen instructions shown;

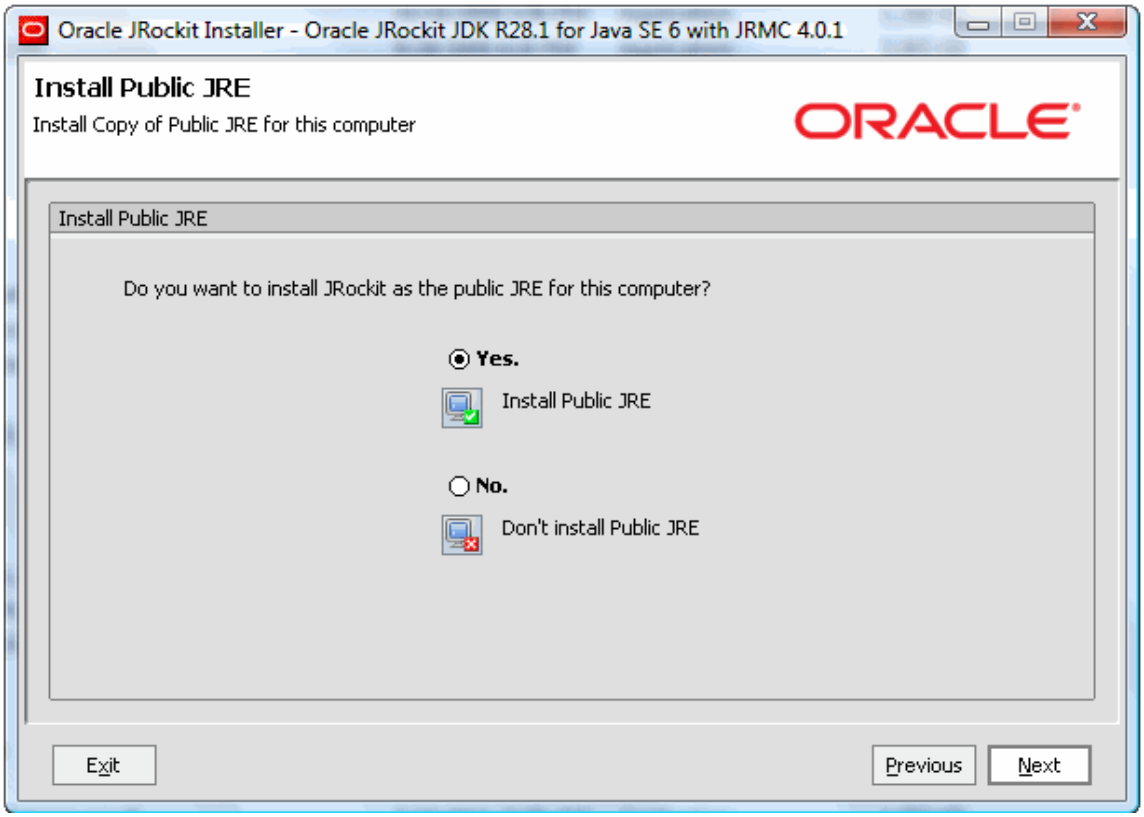

Oracle JRockit Installer - Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
Welcome
This installer will guide you through the installation of
Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
ORACLE
Instructions
Click the Next button to proceed to the next screen. If you want to change entries in a previous screen, click the Previous button. You may quit the installer at any time by clicking the Exit button.
ORACLE
Exit
Previous
Next

Rename the directory where you want to install the product


Oracle JRockit Installer - Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
Choose Product Installation Directories
Provide the directories where you wish to install
Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
ORACLE
Discard Changes
Product Installation Directory
Product Installation Directory
C:\Program Files\Java\jrockit-jdk1.6.0_20-R28.1.0-4.0.1
Browse
Exit
Previous
Next


Oracle JRockit Installer - Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
Optional Components
Please select which of the following optional components you want to install.
ORACLE
Optional Components
Description of Figure B-3 f
ORACLE
Demos and Samples
Source Code
Exit
Previous
Next

Say No


Oracle JRockit Installer - Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
Install Public JRE
Install Copy of Public JRE for this computer
ORACLE
Install Public JRE
Do you want to install JRockit as the public JRE for this computer?
Yes.
Install Public JRE
No.
Don't install Public JRE
Exit
Previous
Next

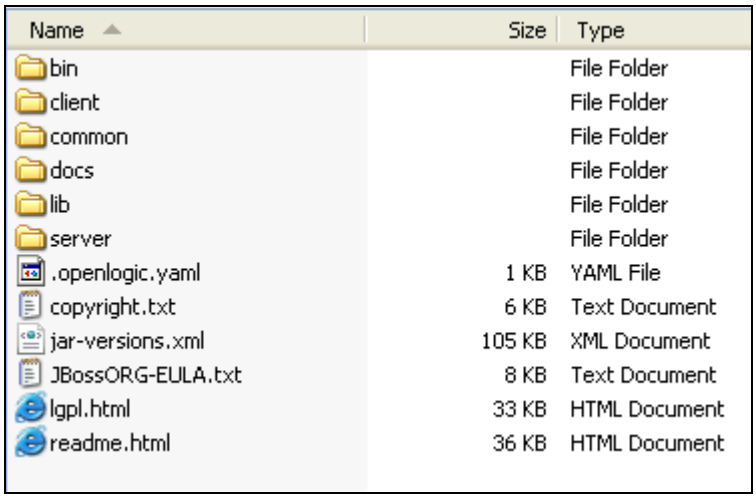
The Oracle JRockit Installer screen appears and the installation process continues, as indicated on the progress meter in the lower right corner of the screen.

Oracle JRockit Installer - Oracle JRockit JDK R28.1 for Java SE 6 with JRMC 4.0.1
Oracle® JRockit Real Time
Gain unparalleled performance with industry-leading Deterministic Garbage Collector
Millisecond response times with "Five Nines" reliability
Improve application performance and latencies with unique tools
Integrated zero-overhead real-time monitoring
Quickly understand and fix performance and latency issues
Benefit from the industry's fastest JVM – Oracle JRockit JVM
ORACLE
Installing JRE to ...C:\Program Files\Java\jrockit-jdk1.6.0_20-R28.1.0-4.0.1\jre
59%
Exit
Previous
Next

When the installation is complete, the Installation Complete screen appears.
Click **Done**.

Appendix J: Installing the JBoss 5.1.0 GA

The JBoss installer can be downloaded from *JBoss* website or the file placed under "C:\RemoteAdmin\Installables\JBoss 5.1.0.GA" can be used. The current version of Jboss Installer will work for Windows XP, Vista and 7. It is in zip format and does not have an installation file. All you have to do is unzip it where ever you want to. In here we will be unzipping in C:\RemoteAdmin\jboss-5.1.0.GA. Once the archive is extracted, JBoss is fully installed and ready for use. You can see the folder structure in the diagram below.



Some Important JBoss Install Directory Structure

The binary distribution unpacks into a top-level JBoss install directory, often referred to as the JBOSS_DIST directory. There are four sub-directories immediately below this:

NOTE: Some directories are dynamically created when you start JBoss. Some directories may not exist in older versions.

bin: contains various scripts and associated files. This is where the run and shutdown scripts, which start and stop JBoss, reside.

client: stores configuration and jar files which may be needed by a Java client application or an external web container. You can select archives as required or use jbossall-client.jar.

docs: contains useful information

lib: jar files which are needed to run the JBoss microkernel. You should never add any of your own jar files here.

server: each of the subdirectories in here is a different Server Configuration. The configuration is selected by passing the option -c <config-name> to the run script.

conf: configuration files including the bootstrap services in jboss-service.xml

lib: static jar files for the services

deploy: services and applications that are hot deployed

data: data files that survive reboot

tmp: temporary files that do not survive reboot

work: work files for Tomcat

default: the default configuration (does not include clustering or corba)

JVM and Memory Settings

Executing *run.conf* in the **bin** directory controls the configuration of JBoss application server. Little change is needed in run.conf. In the file, change **"JAVA_HOME=C:\opt\jdk1.6.0_13" to "JAVA_HOME= C:\jrockit-jdk1.6.0_20-R28.1.0-4.0.1"**. The minimum and maximum memory is set to 128MB and 512MB, respectively.

Creating a JDBC Datasource

Note: Datasource is a name given to the connection set up to a database from a server. The name is commonly used when creating a query to the database.

The syntax for configuring JCA JDBC connection factories has been simplified in 3.2. Rather than configuring the connection manager factory related MBeans via an mbean services deployment descriptor, an abbreviated datasource centric descriptor is used. This is transformed into the standard jboss-service.xml mbean services deployment descriptor using a XSL transform applied by the org.jboss.deployment. XSLSubDeployer included in the jboss-jca.sar deployment. The simplified configuration descriptor is deployed the same as other deployable components. The descriptor must be named using a -ds.xml pattern in order to be recognized by the XSLSubDeployer.

The schema for the datasource descriptor is docs/dtd/jboss-ds_1_0.dtd.

Place the JDBC driver classes in the lib directory of your server. (server/default/lib, for example) Jars placed in the lib directory are NOT hot deployable, so you will need to restart JBoss if it is already running.

Copy the most appropriate sample datasource file into your deploy directory. You can change the name if you like, but the file must end with **-ds.xml** to be recognized as a datasource descriptor.

Edit the datasource definition to match your database connection parameters. The most relevant parameters are:

Jndi-name: The JNDI name under which the Datasource wrapper will be bound. Note that this name is relative to the java:/ prefix. The full JNDI name of the DataSource will

be java:/ + jndi-name. DataSource wrappers are bound under the java:/ prefix since they are not usable outside of the server VM.
Connection-url: The JDBC driver connection URL string for your driver. Make sure that the database name, host and port are correct for the database you are trying to connect to.
Username: This element specifies the default username used when creating a new connection. The actual username may be overridden by the application code getConnection parameters or the connection creation context JAAS Subject.
Password: This element specifies the default password used when creating a new connection. The actual password may be overridden by the application code getConnection parameters or the connection creation context JAAS Subject.

Appendix K: Installing PSTools Suite

You can download the latest version of PSTools Suite from the website or you can find it in C:\RemoteAdmin\Installables\PSTools folder.

In order to install these tools you need to double click each executable file so that it can get registered in the machine. When executing these commands, the application launches itself and installs a Windows service on the remote machine. This Windows service performs the task you requested, using the credentials you provided at launch time. If an end-user UI on the client system is required by the specific tool, this service then launches an additional binary in the user's context. (Remember that services can't throw direct interactive UIs.) The service communicates any necessary information back to the console that launched it. And finally, the service uninstalls itself.

9 References

http://www.mysql.com/about/
http://www.mysql.com/why-mysql/
http://en.wikipedia.org/wiki/MySQL
http://en.wikipedia.org/wiki/JRockit
http://en.wikipedia.org/wiki/Test_case
http://en.wikipedia.org/wiki/Batch_file
http://en.wikipedia.org/wiki/Software_testing
http://en.wikipedia.org/wiki/Windows_service
http://en.wikipedia.org/wiki/User_interface_design
http://en.wikipedia.org/wiki/Remote_administration
http://searchsoa.techtarget.com/definition/WebLogic
http://en.wikipedia.org/wiki/JBoss_application_server
http://en.wikipedia.org/wiki/Oracle_WebLogic_Server
http://technet.microsoft.com/en-us/sysinternals/bb896649
http://www.windowsecurity.com/articles/PsTools-Suite-Part1.html
http://psychology.about.com/od/researchmethods/f/reliabilitydef.htm
http://whatis.techtarget.com/definition/0,289893,sid9_gci213140,00.html
http://technet.microsoft.com/en-us/magazine/2007.03.desktopfiles.aspx
http://tmcpfs.ops.fhwa.dot.gov/cfprojects/uploaded_files/Final%20Q&A.pdf
http://download.oracle.com/docs/cd/E13150_01/jrockit_jvm/jrockit/geninfo/diagnos/understand_jit.html